



**SISTEM INFORMASI PERSEDIAAN BARANG BERBASIS WEB PADA
CV.UNICORNER BANYUWANGI MENGGUNAKAN METODE ECONOMIC ORDER
QUANTITY (EOQ)**

SKRIPSI

oleh

**Ulfa Rahmawati
NIM 102410101043**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER
2018**



**SISTEM INFORMASI PERSEDIAAN BARANG BERBASIS WEB PADA
CV.UNICORNER BANYUWANGI MENGGUNAKAN METODE
ECONOMIC ORDER QUANTITY (EOQ)**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Sistem Informasi (S1)
dan mencapai gelar Sarjana Komputer

oleh

**Ulfa Rahmawati
NIM 102410101043**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER
2018**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang senantiasa memberikan kenikmatan dan kemudahan dalam mengerjakan skripsi ini.
2. Keluarga tercinta Ibu Rukamah, Ayah Suriyono dan Adikku Ahmad Efendi yang selalu mendo'akan dan memberi semangat.
3. Abrian Eko Prasetyo yang selalu memberikan semangat.
4. Teman Seperjuangan di Program Studi Sistem Informasi, Hadi, Cindy, Lukman, Ratna, Iyan dan Bekti.
5. Adik tingkat tercinta Bagus dan Rahmat.
6. Keluarga Besar Program Studi Sistem Informasi;
7. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

“Angin tidak berhembus untuk menggoyangkan pepohonan, melainkan menguji kekuatan akarnya”
(Ali bin Abi Thalib)

“Berjuanglah untuk apa yang kita yakini, tanpa berusaha membuktikan apa pun kepada siapa pun, tetaplah tenang dan tidak banyak cakap, sebagaimana orang yang telah memiliki keberanian untuk menentukan takdirnya sendiri”
(Paulo Coelho)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ulfa Rahmawati

NIM : 102410101043

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Informasi Persediaan Barang Berbasis Web pada CV.UNICORNER Banyuwangi Menggunakan Metode *Economic Order Quantity (EOQ)*”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 17 Januari 2018

Yang menyatakan,

Ulfa Rahmawati

NIM 102410101043

SKRIPSI

**SISTEM INFORMASI PERSEDIAAN BARANG BERBASIS WEB PADA
CV.UNICORNER BANYUWANGI MENGGUNAKAN METODE ECONOMIC ORDER
QUANTITY (EOQ)**

oleh

Ulfa Rahmawati
NIM 102410101043

Pembimbing

Dosen Pembimbing Utama : Prof. Dr. Saiful Bukhori, ST., M.Kom
NIP : 196811131994121001
Dosen Pembimbing Anggota : Windi Eka Yulia Retnani, S.Kom., MT
NIP. 19840305 201012 2 002

PENGESAHAN PEMBIMBING

Sistem Informasi Persediaan Barang Berbasis Web Pada CV.Unicorner Banyuwangi Menggunakan Metode Economic Order Quantity (EOQ) telah diuji dan disahkan pada:

Hari, tanggal : Rabu, 17 Januari 2018

Tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Pembimbing:

Pembimbing I,

Pembimbing II,

Prof. Dr. Saiful Bukhori, ST., M.Kom
NIP 196811131994121001

Windi Eka Yulia Retnani, S.Kom., MT
NIP 198403052010122002

PENGESAHAN

Skripsi berjudul “**Sistem Informasi Persediaan Barang Berbasis Web Pada CV.Unicorner Banyuwangi Menggunakan Metode Economic Order Quantity (EOQ)**”, telah diuji dan disahkan pada:

hari, tanggal : Rabu, 17 Januari 2018

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji:

Penguji I,

Penguji II,

Prof. Drs. Slamin, M.Comp.Sc.,Ph.D
NIP 196704201992011001

M. Arief Hidayat, S.Kom., M.Kom
NIP 198101232010121003

Mengesahkan
Ketua Program Studi,

Prof. Drs. Slamin, M.Comp.Sc.,Ph.D
NIP 196704201992011001

RINGKASAN

Sistem Informasi Persediaan Barang Berbasis Web Pada CV.Unicorner Banyuwangi Menggunakan Metode Economic Order Quantity (EOQ);Ulfa Rahmawati, 102410101043; 2018: 153 halaman; Program Studi Sistem Informasi Universitas Jember.

Pada era globalisasi ini, persaingan di berbagai bidang seperti ekonomi, sosial dan politik semakin tinggi. Khususnya dalam bidang ekonomi, perusahaan dituntut lebih kreatif dalam memenejemen persediaan barang yang diperlukan agar sesuai dengan keinginan pasar dan pihak perusahaan tetap memperoleh keuntungan. Dengan memanfaatkan teknologi informasi, maka penyebaran informasi sangat cepat dan mudah.

CV. Unicorner merupakan sebuah perusahaan yang bergerak dibidang *service* dan penjualan *smartphone*. Dalam mengelola barang dagangan, CV. Unicorner masih menggunakan cara manual tanpa prediksi yang pasti. Hal tersebut memungkinkan adanya kekurangan bahan penumpukan barang secara berlebihan yang berdampak pada kelangsungan toko tersebut. Sehingga diperlukan adanya sebuah sistem untuk perkembangan toko tersebut.

Sistem Informasi Persedian Barang Berbasis Web Menggunakan Metode *Economic Order Quantity* (EOQ) dapat membantu CV. Unicorner dalam menentukan berapa banyak barang yang perlu dipesan dalam periode berikutnya. Metode *Economic Order Quantity* diterapkan pada sistem tersebut karena dapat mengontrol pemesanan ekonomis sehingga CV. Unicorner tidak perlu mengeluarkan banyak biaya dalam setiap pemesanannya.

PRAKATA

Puji syukur ke hadirat Allah SWT atas karena berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi dengan judul “Sistem Informasi Persediaan Barang Berbasis Web Pada CV.Unicorner Banyuwangi Menggunakan Metode Economic Order Quantity (EOQ)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamin, M.CompSc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Dr. Saiful Bukhori, ST., M.Kom, selaku Dosen Pembimbing Utama dan Windy Eka Yulia Retnani, S.Kom., MT., selaku Dosen Pembimbing Anggota yang selalu bijaksana memberikan bimbingan, nasehat serta waktunya selama penulisan skripsi ini;
3. Dr. Saiful Bukhori, ST., M.Kom, selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Semua pihak yang telah banyak membantu penulis dalam menyelesaikan skripsi ini.

Demi perbaikan selanjutnya, saran dan kritik yang membangun akan penulis terima dengan senang hati. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 17 Januari 2018

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	ii
PERSEMBAHAN.....	iii
MOTO.....	iv
PERNYATAAN	Error! Bookmark not defined.
HALAMAN PEMBIMBING.....	Error! Bookmark not defined.i
PENGESAHAN PEMBIMBING	vii
PENGESAHAN	viii
RINGKASAN.....	ix
PRAKATA.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN.....	xvi
BAB 1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan dan Manfaat	2
1.3.1. Tujuan.....	3
1.3.2. Manfaat.....	3
1.4. Ruang Lingkup.....	3
1.5. Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA	5
2.1. CV. Unicorner Banyuwangi.....	5
2.2. Persediaan.....	6

2.2.1. Pengertian Persediaan.....	6
2.2.2. Pentingnya Persediaan.....	7
2.2.3. Peranan Pengendalian Persediaan.....	7
2.3. Economic Order Quantity (EOQ)	8
2.3.1. Frekuensi Pemesanan Selama Setahun.....	11
2.3.2. Jarak Tiap Pesan	11
2.3.3. Persediaan Pengaman (Safety Stock/SS).....	11
2.3.4. Titik Pemesanan Kembali (Re Order Point/ROP).....	12
BAB 3. METODOLOGI PENELITIAN	13
3.1. Tujuan Penelitian.....	13
3.2. Jenis Penelitian.....	13
3.3. Tempat Penelitian.....	13
3.4. Alur Penelitian.....	13
3.4.1. Studi Pustaka	14
3.4.2. Tahap Analisis Masalah dan Pengumpulan Data	14
3.4.3. Tahap Perancangan Sistem.....	15
3.4.4. Tahap Penerapan.....	17
BAB 4. DESAIN DAN PERANCANGAN SISTEM.....	18
4.1. Studi Pustaka	18
4.2. Pengumpulan Data.....	18
4.3. Analisis Kebutuhan	18
4.3.1. Kebutuhan Fungsional.....	19
4.3.2. Kebutuhan Non Fungsional.....	19

4.4.	Desain Sistem.....	19
4.4.1.	<i>Bussines Process</i>	19
4.4.2.	<i>Use case</i> Diagram.....	20
4.4.3.	<i>Use case</i> Skenario	23
4.4.4.	<i>Activity</i> Diagram.....	26
4.4.5.	<i>Sequence</i> Diagram.....	28
4.4.6.	<i>Class</i> Diagram	29
4.4.7.	<i>Entity Relationship Diagrams</i> (ERD)	30
4.5.	Penulisan Kode Program.....	31
4.6.	Pengujian Program	31
4.6.1.	Pengujian <i>White box</i>	32
4.6.2.	Pengujian <i>Black box</i>	35
4.7.	Pemeliharaan.....	35
BAB 5.	HASIL DAN PEMBAHASAN	36
5.1.	Pembahasan	36
5.1.1.	Pengumpulan Dataset	36
5.1.2.	Penggunaan Metode Economic Order Quantity	36
5.2.	Hasil Implementasi Sistem Persediaan Barang.....	39
BAB 6.	PENUTUP	46
6.1	Kesimpulan.....	46
6.2	Saran.....	46
DAFTAR PUSTAKA	47

DAFTAR TABEL

	Halaman
Tabel 4.1 Definisi Aktor	21
Tabel 4.2 Definisi <i>Use case</i>	22
Tabel 4.3. <i>Use case</i> Skenario Manajemen Data EOQ	26
Tabel 4.4. Manajemen EOQ Jalur 1.....	35
Tabel 4.5. Manajemen EOQ Jalur 2.....	35
Tabel 4.6. Manajemen EOQ Jalur 3.....	35
Tabel 4.7. Manajemen EOQ Jalur 4.....	35
Tabel 4.8. Manajemen EOQ Jalur 5.....	35
Tabel 4.9. Manajemen EOQ Jalur 6.....	35
Tabel 5.1. Penjualan Flashdisk Tahun 2014	38

DAFTAR GAMBAR

	Halaman
Gambar 0.1. Flowchart EOQ	11
Gambar 0.2. Alur Penelitian	ix
Gambar 0.3. Model <i>Increment</i>	16
Gambar 0.4. <i>Bussines Process</i>	21
Gambar 0.5. <i>Use case</i> Diagram Sistem Informasi Persediaan Barang.....	24
Gambar 0.6. <i>Activity Diagram</i> Manajemen Data EOQ_	28
Gambar 0.7. <i>Sequence</i> Manajemen Data EOQ	30
Gambar 0.8. <i>Class</i> Diagram Sistem Informasi Persediaan Barang	31
Gambar 4.6. ERD Sistem Informasi Persediaan Barang	32
Gambar 4.7. Diagram Alir Manajemen EOQ.....	33
Gambar 0.9. Tampilan <i>Login</i>	38
Gambar 0.10. Tampilan halaman <i>User</i>	39
Gambar 5.3. Tampilan form tambah <i>User</i>	39
Gambar 5.4. Tampilan halaman Pembelian.....	40
Gambar 5.5. Tampilan form tambah Pembelian.....	40
Gambar 5.6. Tampilan halaman data barang	41
Gambar 5.7. Tampilan halaman tambah barang	41
Gambar 5.8. Tampilan halaman data Merk.....	42
Gambar 5.9. Tampilan tambah data Merk	42
Gambar 5.10. Tampilan halaman Kategori.....	43
Gambar 5.11. Tampilan tambah data Kategori.....	43
Gambar 5.12. Tampilan halaman data penjualan.....	44
Gambar 5.13. Tampilan halaman manajemen EOQ.....	44

DAFTAR LAMPIRAN

LAMPIRAN A. <i>USE CASE</i> SKENARIO.....	49
A.1. <i>Use case</i> Skenario Autentikasi (<i>Login</i>).....	49
A.2. <i>Use case</i> Skenario Manajemen Data <i>User</i>	50
A.3. <i>Use case</i> Skenario Manajemen Data Pembelian.....	51
A.4. <i>Use case</i> Skenario Manajemen Data Barang.....	53
A.5. <i>Use case</i> Skenario Manajemen Data Merk_	55
A.6. <i>Use case</i> Skenario Manajemen Data Kategori.....	56
A.7. <i>Use case</i> Skenario Manajemen Data Penjualan	58
A.8. <i>Use case</i> Skenario Melihat Data Transaksi Barang	60
LAMPIRAN B. <i>ACTIVITY</i> DIAGRAM.....	61
B.1. Activity diagram Autentikasi(<i>login</i>)	61
B.2. Activity diagram manajemen data <i>user</i>	62
B.3. Activity diagram manajemen data pembelian.....	63
B.4. Activity diagram manajemen data barang.....	64
B.5. Activity diagram manajemen data merk.....	65
B.6. Activity diagram manajemen data kategori.....	66
B.7. Activity diagram manajemen data penjualan	67
B.8. Activity diagram melihat data transaksi barang	68
LAMPIRAN C. <i>SEQUENCE</i> DIAGRAM	69
C.1. <i>Sequence</i> Diagram <i>Login</i>	69
C.2. <i>Sequence</i> Diagram Manajemen Data <i>User</i>	69
C.3. <i>Sequence</i> Diagram Manajemen Data Pembelian.....	70
C.4. <i>Sequence</i> Diagram Manajemen Data Barang.....	70
C.5. <i>Sequence</i> Diagram Manajemen Data Merk.....	71
C.6. <i>Sequence</i> Diagram Manajemen Data Kategori	71
C.7. <i>Sequence</i> Diagram Manajemen Data Penjualan.....	72
C.8. <i>Sequence</i> Diagram Melihat Data Transaksi Barang.....	72

LAMPIRAN D. PENULISAN KODE PROGRAM.....	73
D.1. Autentikasi (<i>login</i>)	73
D.2. Autentikasi (<i>logout</i>)	73
D.3. Manajemen Data <i>User</i>	74
D.4. Manajemen Data Pembelian	75
D.5. Manajemen Data Barang.....	76
D.6. Manajemen Data Merk.....	78
D.7. Manajemen Data Kategori	79
D.8. Manajemen Data Penjualan.....	80
D.9. <i>view User</i> oleh <i>Owner</i>	80
D.10. Data Transaksi <i>view</i> Barang.....	80
D.11. Data Transaksi <i>view</i> Pembelian	80
LAMPIRAN E. PENGUJIAN <i>WHITE BOX</i>	81
E.1. Autentikasi (<i>login</i>).....	81
E.2. <i>Logout</i>	84
E.3. Menu <i>User</i>	85
E.4. Tambah <i>User</i>	86
E.5. <i>Insert User</i>	87
E.6. <i>Edit User</i>	88
E.7. <i>Update User</i>	89
E.8. Hapus <i>User</i>	90
E.9. Menu Pembelian.....	91
E.10. Tambah Pembelian	92
E.11. <i>Insert Pembelian</i>	93
E.12. <i>Edit Pembelian</i>	95
E.13. <i>Update Pembelian</i>	96
E.14. Hapus Pembelian	97
E.15. Menu Barang	98

E.16. Tambah Barang	99
E.17. <i>Insert</i> Barang	100
E.18. <i>Edit</i> Barang	101
E.19. <i>Update</i> Barang	102
E.20. Hapus Barang	103
E.21. Menu Merk	104
E.22. Tambah Merk	105
E.23. <i>Insert</i> Merk	106
E.24. Hapus Merk	107
E.25. Menu Kategori	108
E.26. Tambah Kategori	109
E.27. <i>Insert</i> Kategori	110
E.28. Hapus Kategori	112
E.29. Menu Penjualan	113
E.30. Tambah Penjualan	113
E.31. <i>Insert</i> Penjualan	114
E.32. Hapus Penjualan	115
E.33. Transaksi Barang	116
LAMPIRAN F. PENGUJIAN <i>BLACK BOX</i>	118
F.1. Autentikasi (<i>login</i>)	118
F.2. Manajemen Data <i>User</i>	118
F.3. Manajemen Data Pembelian	119
F.4. Manajemen Data Barang	121
F.5. Manajemen Data Merk	122
F.6. Manajemen Data Kategori	123
F.7. Manajemen Data Penjualan	124
F.8. Manajemen Data Transaksi	125
LAMPIRAN G. DATASET	126

Lampiran G. Data penjualan barang di CV.Unicorner 126



BAB 1. PENDAHULUAN

Bab ini merupakan awal untuk penulisan skripsi. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan skripsi.

1.1. Latar Belakang

CV.Unicorner adalah sebuah perusahaan dalam bidang pelayanan jasa *service* dan penjualan *smartphone*. Didirikan pada tahun 2008 di kota jember. Berawal dari nama blackberry corner yang mana pelayanan jasa *service* hanya mencakup *smartphone* dengan merk blackberry. Kemudian pada tahun 2012 pemilik CV. Unicorner membuka cabang gerai di Banyuwangi dengan nama yang sama yakni Blackberry Corner. Kemudian seiring perkembangan teknologi dan makin banyak juga *smartphone* yang beredar dipasaran, sehingga CV. Unicorner mulai merambah hingga lebih luas lagi dalam bidang pelayanan jasa *service* alat telekomunikasi berbasis *smartphone*. Hal inilah yang mendorong nama gerai Blackberry Corner menjadi Techno Corner.

Selama ini segala aktivitas di CV. Unicorner baik transaksi penjualan, pembelian dan kegiatan lainnya masih dilakukan secara manual. Oleh karena itu pengelola dan karyawan disana sering mengalami kesulitan dalam mengontrol stok barang karena memungkinkan adanya kelebihan pada satu jenis barang dan kekurangan pada barang lain. Pada tahun 2014 CV. Unicorner mengalami peningkatan pelanggan sehingga sering kehabisan stok, jumlah permintaan seluruh barang dalam satu tahun 547 unit sedangkan persediaan toko saat itu hanya 500 unit. Pihak toko harus melakukan pemesanan dadakan yang memakan banyak waktu dan mengakibatkan pelanggan harus menunggu sampai barang tersedia. Hal itu menyebabkan sering terjadinya kekecewaan pelanggan karena barang yang akan dibeli ternyata tidak tersedia sehingga dapat menyebabkan penurunan pemasukan. Alternative lain untuk menutupi kekurangan barang yaitu dengan membeli barang

yang dibutuhkan dari toko lain yang harganya jauh lebih tinggi dari harga pusat, akibatnya pihak Unicorner memperoleh sedikit keuntungan bahkan mengalami kerugian.

Pengelola dan karyawan di CV. Unicorner melakukan pemesanan dengan memperkirakan jumlah barang tanpa melakukan penghitungan yang pasti. Sehingga cara tersebut dinilai kurang efisien dalam menangani masalah pengontrolan stok barang. Oleh karenanya diperlukan adanya sebuah metode untuk meramalkan berapa banyak barang yang dibutuhkan di toko tersebut, supaya tidak terjadi persediaan barang yang menumpuk maupun kekurangan stok. Dalam hal ini, metode yang cocok untuk diterapkan di toko tersebut ialah metode *Economic Order Quantity* yang merupakan salah satu metode untuk membantu menangani masalah persediaan. Pada dasarnya metode ini berusaha mencari jawaban optimal dalam menentukan jumlah pesanan yang terbaik.

Pada penelitian ini akan dibuat sistem berbasis web untuk manajemen persediaan pada CV. Unicorner Banyuwangi menggunakan metode *Economic Order Quantity (EOQ)*. Metode *Economic Order Quantity (EOQ)* dipilih karena lebih efisien dibandingkan dengan metode yang lain. Perancangan sistem ini diharapkan dapat membantu pengelola dan karyawan CV. Unicorner dalam menangani masalah yang ada.

1.2. Perumusan Masalah

Berdasarkan latar belakang diatas,dapat dirumuskan beberapa permasalahan antara lain :

1. Bagaimana mendapatkan persediaan barang yang efektif?
2. Bagaimana membangun sistem persediaan barang berbasis web pada CV. Unicorner Banyuwangi dengan menggunakan metode EOQ?

1.3. Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang diinginkan.

dari penelitian ini.

1.3.1. Tujuan

Adapun tujuan yang ingin dicapai dari penelitian ini adalah sebagai berikut:

1. Mendapatkan pengelolaan stok barang yang efektif.
2. Membangun sistem informasi persediaan barang berbasis web pada CV. Unicorner Banyuwangi menggunakan metode EOQ.

1.3.2. Manfaat

Manfaat dari penelitian ini adalah:

1. Bagi CV.Unicorner

Dengan dibangunnya sistem informasi pengelolaan stok barang menggunakan metode EOQ ini dapat mempermudah CV.Unicorner dalam menganalisa banyaknya barang yang dibutuhkan sehingga dapat memangkas biaya penyimpanan barang akibat stok yang terlalu besar.

2. Bagi Penulis

Dapat menambah ilmu pengetahuan dalam mengembangkan teori dan penerapannya pada dunia praktik, khususnya tentang Sistem informasi pengelolaan stok barang dan sebagai media dalam menyelesaikan Tugas Akhir untuk jenjang S-1 pada Program Studi Sistem Informasi.

1.4. Ruang Lingkup

Aplikasi yang dibangun adalah Sistem Informasi Persediaan Barang Berbasis Web Pada CV.Unicorner Banyuwangi Menggunakan Metode *Economic Order Quantity* (EOQ), agar dapat membantu pihak Unicorner dalam menekan biaya sehingga mendapat keuntungan yang maksimal. Ruang lingkup dari penelitian ini adalah:

1. Sistem informasi ini tidak sampai pada tahap penghitungan keuangan

2. Sistem yang dibangun berbasis web dan menggunakan konsep *Object Oriented Programming* (OOP).
3. Sistem digunakan untuk membantu CV. Unicorner dalam mengelola persediaan barang.
4. Metode *Economic Order Quantity* (EOQ) digunakan untuk penghitungan data barang.
5. Database yang digunakan adalah *My Structure Query Language* (MySQL).

1.5. Sistematika Penulisan

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut:

1. Pendahuluan
Bab ini menjelaskan latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan.
2. Tinjauan Pustaka
Bab ini berisi materi, informasi, kajian teori dan studi terdahulu yang digunakan dalam penulisan.
3. Metode Penulisan
Bab ini menjelaskan tentang metode penulisan yang digunakan selama proses penulisan berlangsung.
4. Analisis dan Perancangan Sistem
Bab ini menjelaskan tentang analisis dan perancangan sistem yang dikembangkan.
5. Hasil dan Pembahasan
Bab ini menjelaskan tentang hasil dan pembahasan dari sistem yang sudah dibuat.
6. Penutup
Bab ini berisi tentang kesimpulan dari penulis dan saran untuk penulis selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Bab ini menjelaskan secara lengkap tentang profil objek penulisan, teori – teori yang ada dan informasi yang mendukung dalam penulisan.

2.1. CV. Unicorner Banyuwangi

CV.Unicorner adalah sebuah perusahaan dalam bidang pelayanan jasa service dan penjualan *smartphone*. Didirikan pada tahun 2008 di kota jember. Berawal dari nama blackberry corner yang mana pelayanan jasa service hanya mencakup *smartphone* dengan merk blackberry. Kemudian pada tahun 2012 pemilik CV. Unicorner membuka cabang gerai di Banyuwangi dengan nama yang sama yakni Blackberry Corner.

Setelah itu dalam perkembangannya *techno corner* merambah dunia yang lebih luas, yakni penjualan unit *smartphone*, aksesoris *handphone*, dan sparepart *handphone*. Dalam penjualan ini *techno corner* memberikan pelayanan lebih terhadap pelanggan. Sehingga ketika pelanggan membeli unit *smartphone* di *techno corner*, pelanggan tetap mendapatkan garansi berupa konsultasi gratis, hingga pelayanan maintenance lainnya atas unit yang telah dibeli. Sehingga dengan begitu, pelanggan tetap merasa nyaman ketika membeli unit *smartphone* di *techno corner*. Karena tidak akan merasa khawatir ketika ada hal – hal yang terjadi pada unit yang dibelinya.

Gerai-gerai Techno Corner berada di 2 (dua) lokasi, dalam 2 (dua) kota yang berbeda. Yakni di Jl. Gajah Mada no.2 Banyuwangi dan di Jl Sumatra 82A Jember. Di Techno Corner melayani jasa service untuk berbagai merk *smartphone* yang banyak beredar dipasaran, seperti Blacberry, Samsung, Sony, LG, OPPO, Apple, Advan, Evercoss, Nokia Lumia, dll.

Visi perusahaan *techno corner* dalam memberikan pelayanan terhadap pelanggan adalah sebagai berikut :

1. Pelanggan mendapat garansi atas apa yang telah dikerjakan oleh *techno corner*.

2. Pelanggan merasa puas atas apa yang dikerjakan oleh techno corner
3. Pelanggan tidak merasa khawatir saat terjadi sesuatu pada unit yang dibeli, khususnya yang dibeli di techno corner.

Dengan adanya visi diatas maka CV. Unicorner memiliki misi yaitu seluruh crew Techno Corner memberikan pelayanan terbaik pada pelanggan yang datang. Baik yang membeli unit ataupun hanya sekedar ingin konsultasi atas apa yang terjadi pada smartphone mereka. Pegawai memberikan garansi atas apa yang telah dikerjakan pada unit pelanggan.

2.2 Persediaan

2.2.1 Pengertian Persediaan

Persediaan sangat penting bagi kelangsungan sebuah perusahaan, baik perusahaan besar maupun kecil. Adapun definisi persediaan menurut beberapa ahli sebagai berikut :

- a. Menurut R. Agus Sartono (2010), persediaan pada umumnya merupakan salah satu jenis aktiva lancar yang jumlahnya cukup besar dalam suatu perusahaan. Ditinjau dari segi neraca persediaan adalah barang-barang atau bahan yang masih tersisa pada tanggal neraca, atau barang-barang yang akan segera dijual, digunakan atau diproses dalam periode normal perusahaan (Sartono, 2010).
- b. Menurut Ristono (2009), Persediaan dapat diartikan sebagai barang-barang yang disimpan untuk digunakan atau dijual pada masa atau periode yang akan datang. Persediaan terdiri dari persediaan bahan baku, persediaan bahan setengah jadi dan persediaan barang jadi. Persediaan bahan baku dan bahan setengah jadi disimpan sebelum digunakan atau dimasukkan kedalam proses produksi, sedangkan persediaan barang jadi atau barang dagangan disimpan sebelum dijual atau dipasarkan. Dengan demikian setiap perusahaan yang melakukan kegiatan usaha umumnya memiliki persediaan (Ristono, 2009).
- c. Persediaan menurut Sofjan Assauri (2004), adalah suatu aktiva yang meliputi barang-barang milik perusahaan yang dimaksud untuk dijual dalam satu periode

usaha yang normal atau persediaan barang baku yang menunggu penggunaannya dalam suatu proses produksi (Assauri, 2004).

Dari definisi tersebut dapat disimpulkan bahwa persediaan merupakan barang-barang yang disediakan oleh perusahaan untuk kemudian dijual ke konsumen untuk memperoleh pendapatan.

2.2.2 Pentingnya Persediaan

Persediaan merupakan hal yang perlu diperhatikan dalam bisnis perdagangan. Karena dapat mempermudah jalannya proses pengembangan sebuah perusahaan dalam menangani penjualan produk tertentu yang memerlukan pemesanan secara berturut-turut. Selain itu persediaan juga berpengaruh untuk meningkatkan kemajuan perusahaan, dengan selalu mengupdate kebutuhan masyarakat maka perusahaan tersebut mampu mengatasi persaingan pasar. Dengan begitu ekonomi perusahaan juga terselamatkan.

2.2.3 Peranan Pengendalian Persediaan

Menurut pendapat Assauri (2004) pengendalian persediaan adalah salah satu kegiatan dari urutan kegiatan-kegiatan yang bertautan erat satu sama lain dalam seluruh operasi produksi perusahaan tersebut sesuai dengan apa yang telah direncanakan lebih dahulu baik waktu, jumlah, kualitas maupun biayanya (Assauri, 2004). Pengendalian persediaan yang dijalankan suatu perusahaan selalu mempunyai tujuan tertentu, dari itu perusahaan harus mengadakan persediaan yang tepat. Karena apabila persediaan terlalu berlebihan akan banyak biaya yang dikeluarkan untuk modal dan memelihara persediaan tersebut dan bila persediaan terlalu kecil akan mengganggu kelancaran dari kegiatan penjualan. Menurut Assauri (2004), tujuan pengendalian persediaan secara terinci dapat dinyatakan sebagai usaha untuk :

- a. Menjaga jangan sampai perusahaan kehabisan persediaan sehingga dapat mengakibatkan terhentinya kegiatan produksi.

- b. Menjaga agar pembentukan persediaan oleh perusahaan tidak terlalu besar atau berlebih-lebihan.
- c. Menjaga agar pembelian secara kecil-kecilan dapat dihindari karena ini akan berakibat biaya pemesanan terlalu besar (Assauri, 2004).

Dari keterangan diatas dapat disimpulkan bahwa tujuan pengendalian persediaan yaitu untuk memperoleh kualitas dan jumlah yang tepat dari barang-barang yang tersedia pada waktu yang dibutuhkan dengan biaya-biaya yang optimal untuk keuntungan perusahaan.

2.3 *Economic Order Quantity (EOQ)*

Salah satu metode manajemen persediaan yang paling terkenal adalah metode *Economic Order Quantity* atau biasa disebut dengan EOQ. Metode ini dapat digunakan baik untuk barang yang dibeli maupun untuk barang yang diproduksi sendiri. Dalam penelitian ini penulis menggunakan penelitian terdahulu untuk mempermudah penulis dalam menyusun penelitian ini. Beberapa penelitian tersebut antara lain.

- a. Penelitian yang dilakukan oleh Faqih Abidin (2011), dengan judul Sistem Informasi Penjualan dan Pengontrolan Stok Barang Menggunakan Metode EOQ (*Economic Order Quantity*) Studi Kasus di Bengkel Sukasari Motor tujuan penelitian tersebut yaitu mempermudah dalam mengontrol ketersediaan barang di bengkel tersebut. Dari hasil penelitiannya disebutkan bahwa ketika frekuensi pemesanan 1 kali, maka total biaya sangat mahal (185.000,-), demikian seterusnya bila pemesanan dilakukan sebanyak 2 kali, 3 kali dan seterusnya. Begitupun ketika pemesanan dilakukan diatas 6 kali pemesanan, biaya total yang harus dikeluarkanpun meningkat kembali. Dari uji coba tersebut terlihat bahwa ketika pemesanan dilakukan sebanyak 6 kali (sesuai dengan hasil perhitungan EOQ), total biayanya paling rendah dibandingkan diatas 6 kali ataupun dibawah 6 kali pemesanan.
- b. Selanjutnya penelitian dari Ginting (2013), dengan judul Perancangan

Sistem Pengontrolan Stok Barang Pada Blesscom Komputer dengan Metode EOQ. Dengan hasil penelitian perbandingan pengontrolan dan juga pengorderan dengan menggunakan metode EOQ lebih hemat, biaya pengorderan sebelumnya sebesar Rp.60000 sedangkan menggunakan EOQ yakni sebesar $8 \times \text{Rp.5000} = \text{Rp.40000}$ jadi selisih Rp.20000 dalam pengorderan satu jenis barang, apabila terjadi dalam banyak jenis barang maka keefisienan yang didapatkan semakin besar sehingga pihak perusahaan dapat menambah modal untuk pembelian barang-barang yang lain (Ginting, 2013).

Dari penelitian-penelitian diatas dapat disimpulkan bahwa dengan menerapkan metode EOQ, persediaan barang lebih optimal. Karena perusahaan ini merupakan perusahaan yang bergerak di bidang penjualan, maka metode EOQ sangat cocok diterapkan untuk mengatasi permasalahan di perusahaan tersebut.

Metode EOQ mengasumsikan permintaan secara pasti dengan pemesanan yang dibuat secara konstan serta tidak adanya kekurangan persediaan. Adapun asumsi yang harus dipenuhi dalam metode EOQ, yaitu:

1. Tingkat permintaan datang secara konstan, berulang-ulang dan diketahui.
2. Tidak diperbolehkan terjadinya kehabisan persediaan.
3. Bahan yang dipesan dan diproduksi pada satu waktu.
4. Biaya pemesanan setiap unit adalah konstan.
5. Barang yang dipesan tunggal (Rangkuti, 2004).

Tetapi dalam kenyataannya asumsi-asumsi di atas tidak dapat dipenuhi semuanya, karena kondisi dan keadaan yang terkadang bisa terjadi tiba-tiba. Oleh karena itu metode EOQ mengalami pengembangan yang disesuaikan dengan kondisi dan keadaan dari perusahaan itu sendiri. Secara umum metode EOQ dapat dilihat pada persamaan 1:

$$Q = \sqrt{\frac{2DS}{H}} \dots\dots\dots(1)$$

Keterangan :

Q = Jumlah barang yang disarankan untuk dipesan.

D = Jumlah kebutuhan dalam satu tahun, dihitung dari total penjualan tahun sebelumnya dikurangi dengan retur penjualan yang disebabkan karena barang tidak sesuai dengan permintaan pelanggan tahun sebelumnya dari barang yang akan dihitung EOQ nya.

S = Biaya pesan, dihitung dari berapa besar biaya yang keluar untuk setiap kali melakukan pemesanan, meliputi biaya surat menyurat, biaya telpon, transportasi, dan sebagainya.

H = Biaya penyimpanan, persentase biaya penyimpanan yaitu total asset dibagi dengan biaya penyimpanan yang meliputi biaya sewa tempat, biaya asuransi jika ada.

Langkah-langkah dalam penghitungan EOQ bisa dilihat pada gambar 2.1.



Gambar 0.1. Flowchart EOQ

Penjelasan gambar 2.1, gambar tersebut merupakan penjelasan mengenai alur program persediaan barang menggunakan metode EOQ (*Economic Order Quantity*). Untuk memperoleh data informasi EOQ (*Economic Order Quantity*) langkah yang harus dilakukan yaitu, pilih item barang yang akan dihitung, input jumlah kebutuhan dalam satu tahun, dihitung dari total penjualan sebelumnya. Lalu input biaya pemesanan dan penyimpanan, kemudian dilakukan penghitungan EOQ oleh sistem hingga mendapat informasi pemesanan barang secara optimal.

2.3.1. Frekuensi Pemesanan Selama Setahun

Dapat dilihat pada persamaan 2:

$$F = \frac{D}{Q} \dots\dots\dots(2)$$

Keterangan :

F = Frekuensi pemesanan

D = Jumlah kebutuhan dalam satu tahun

Q = Jumlah pemesanan EOQ (*Economic Order Quantity*)

2.3.2. Jarak Tiap Pesan

Dapat dilihat pada persamaan 3:

$$T = \frac{365}{F} \dots\dots\dots(3)$$

Keterangan :

T = Jarak tiap pesan

F = Frekuensi pemesanan

2.3.3. Persediaan Pengaman (*Safety Stok/SS*)

Pengertian persediaan pengaman (*Safety Stock*) menurut Ranguti (2004:10) adalah persediaan tambahan yang diadakan untuk melindungi atau menjaga kemungkinan terjadinya kekurangan bahan (*Stock Out*).

Secara umum dapat dilihat pada persamaan 4:

$$SS = (\text{Pemakaian Maksimum} - \text{Pemakaian Rata-Rata}) L \dots\dots\dots(4)$$

Keterangan:

$SS = \text{Safety Stock}$

$L = \text{Lead Time}$

2.3.4. Titik Pemesanan Kembali (*Reorder Point/ROP*)

Selain memperhitungkan konsep EOQ (*Economic Order Quantity*), perusahaan juga perlu memperhitungkan kapan harus dilakukan pemesanan kembali (*Re Order Point*).

Pengertian *Re Order Point* (ROP) menurut Rangkuti (2004:83) adalah strategi operasi persediaan merupakan titik pemesanan yang harus dilakukan suatu perusahaan sehubungan dengan adanya *Lead Time* dan *Safety Stock*. *Lead Time* adalah lamanya waktu barang yang dipesan hingga sampai di perusahaan.

Sedangkan menurut Riyanto (2001:83) ROP adalah saat atau titik dimana harus diadakan pesanan lagi sedemikian rupa sehingga kedatangan atau penerimaan material yang dipesan adalah tepat waktu dimana persediaan diatas *Safety Stock* sama dengan nol.

Dapat dilihat pada persamaan 5:

$$ROP = (d \times L) + SS \dots\dots\dots(5)$$

Keterangan:

$ROP = \text{Re Order Point}$

$d = \text{Jumlah permintaan (per unit) dalam waktu yang ditentukan. Dengan cara membagi permintaan tahunan dengan jumlah periode yang digunakan.}$

$L = \text{Lead Time}$

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang tujuan penelitian, jenis penelitian, tempat penelitian, dan alur penelitian yang akan digunakan dalam pembuatan Sistem Informasi Persediaan Barang Berbasis Web Pada CV.Unicorner Banyuwangi Menggunakan Metode *Economic Order Quantity* (EOQ).

3.1. Tujuan Penelitian

Dari latar belakang yang sudah diuraikan, tujuan dari penelitian ini adalah membuat Sistem Informasi Persediaan Barang Berbasis Web Pada CV.Unicorner Banyuwangi Menggunakan Metode *Economic Order Quantity* (EOQ) untuk menekan biaya agar tidak terjadi penumpukan stok barang yang berlebihan.

3.2. Jenis Penelitian

Penelitian ini dilakukan dengan jenis metode kualitatif yaitu dengan memusatkan kendala-kendala apa saja yang terjadi di perusahaan tersebut. Sehingga dapat disimpulkan secara deskriptif.

3.3. Tempat Penelitian

Penelitian dilakukan di CV. Unicorner Jl. Gajah Mada no.2 Banyuwangi.

3.4. Alur Penelitian

Metode penelitian yang akan digunakan dalam pembuatan Sistem Informasi Persediaan Barang Berbasis Web Pada CV.Unicorner Banyuwangi Menggunakan Metode *Economic Order Quantity* (EOQ) diawali dengan studi pustaka, analisis masalah dan pengumpulan data, perancangan sistem, dan penerapan. Model SDLC yang digunakan adalah model inkremental. Alur penelitian dapat dilihat pada gambar 3.1.



Gambar 0.1. Alur Penelitian

3.4.1. Studi Pustaka

Tahap awal yang dilakukan yaitu studi pustaka, sebagai penunjang bagi penulis untuk memahami materi dan konsep supaya mudah dalam mengerjakan tugas akhir. Dengan mengumpulkan data dari buku-buku referensi, skripsi, dan jurnal *online* dari penelitian terdahulu yang berkaitan dengan penerapan metode EOQ pada pengelolaan stok barang.

3.4.2. Tahap Analisis Masalah dan Pengumpulan Data

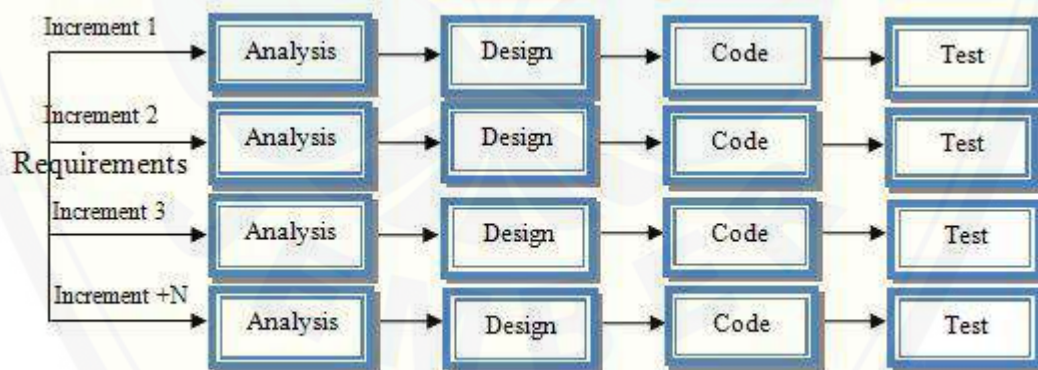
Pada tahap analisis ini, penulis mengidentifikasi masalah yang ada pada CV.Unicorner dengan mengumpulkan data-data melalui wawancara pada pihak yang

bersangkutan dan observasi. Pada tahapan ini diketahui bahwa proses pengelolaan barang di Unicorner kurang optimal. Sehingga sering terjadi penumpukan barang yang mengakibatkan pihak toko mengalami kerugian.

3.4.3. Tahap Perancangan Sistem

Tahap perancangan sistem ini dibuat agar nantinya pembuatan sistem lebih mudah dan tertata. Untuk Sistem Informasi Persediaan Barang Berbasis Web pada CV.UNICORNER Banyuwangi penulis menerapkan model *increment* sebagai salah satu dari model SDLC (*Software Development Life Cycle*) atau sering disebut juga *System Development Life Cycle*.

Seperti yang dikatakan Rosa dan Shalahuddin (2014), bahwa model inkremental dibuat untuk mengatasi kelemahan dari model air terjun yang tidak mengakomodasi iterasi, dan mengatasi kelemahan dari model prototipe yang memiliki proses terlalu pendek dan setiap iteratif prosesnya tidak selalu menghasilkan produk (bisa jadi hanya prototipe). Model inkremental menghasilkan produk/aplikasi untuk setiap tahapan inkremen (S. Rosa A dan Salahuddin, 2014).



Gambar 0.2. Model *Increment*

Model increment memiliki beberapa tahapan yaitu:

1. Requirements (Penentuan Kebutuhan)

Penentuan kebutuhan, dimana pelanggan dan pengembang sama-sama

mendefinisikan kebutuhan dan garis besar sistem yang akan dibuat.

2. Analysis (Analisis)

Tahap dimana pengembang menganalisis semua kebutuhan yang telah didapat. Tahap analisis juga dilakukan untuk menentukan pekerjaan yang akan dilakukan terlebih dahulu dalam pembuatan sistem.

3. Design (Desain sistem)

Merupakan tahap dimana pengembang melakukan perancangan software yang terbuka agar dapat diterapkan sistem pembangunan perbagian pada tahapan selanjutnya. Sistem ini berbasis *Object Oriented Programming* maka dibuatlah diagram-diagram untuk mempermudah dalam mengubah kebutuhan-kebutuhan dan mempermudah proses pengkodean. Diagram-diagram tersebut antara lain:

- a. *Bussiness Process*
- b. *Usecase Diagram*
- c. *Sequence Diagram*
- d. *Activity Diagram*
- e. *Class Diagram*
- f. *Entity Relationship Diagram (ERD)*

4. Code (Penulisan kode program)

Setelah melakukan proses desain selanjutnya ada pengkodean, dimana pengembang menerjemahkan desain tersebut kedalam bahasa yang dimengerti oleh komputer. Sistem ini berbasis web sehingga menggunakan PHP (*Page Hyper Text Pre-Processor*). Sedangkan untuk database, program ini menggunakan MySQL.

5. Test (Pengujian program)

Merupakan tahap pengujian, pengujian sistem dilakukan dengan menggunakan dua metode, yaitu *white box* dan *black box*.

- a. Pengujian *white box* merupakan pengujian yang terfokus terhadap struktur kontrol program, memastikan bahwa semua *statement* pada program telah dieksekusi paling tidak satu kali selama pengujian dan semua kondisi logis telah diuji. Teknik pengujian *white box* menggunakan *Cyclomatic Complexity* untuk

melakukan serangkaian pengujian yang *independent* secara linear. *Cyclomatic Complexity* merupakan suatu sistem pengukuran yang menyediakan ukuran kuantitatif dari kompleksitas logika suatu program. Hasil dari *Cyclomatic Complexity* digunakan untuk menentukan banyaknya *independent path*. *Independent path* adalah sebuah kondisi pada program yang menghubungkan *node* awal dengan *node* akhir. Pengujian ini dilakukan oleh pihak pengembang.

Penghitungan *Cyclomatic Complexity*, yaitu:

$$V(G) = E - N + 2$$

Keterangan:

$V(G)$ = *Cyclomatic Complexity* untuk *flow graph* G

E = Jumlah *edge* (panah)

N = Jumlah *node* (lingkaran)

- b. *Black box testing* merupakan sebuah metode yang digunakan untuk menemukan kesalahan dan mendemonstrasikan fungsional aplikasi saat dioperasikan, apakah *input* diterima dengan benar dan *output* yang dihasilkan telah sesuai dengan yang diharapkan. Pengujian ini dilakukan oleh pelanggan.

3.4.4. Tahap Penerapan

Tahap penerapan ini merupakan tahap dimana aplikasi yang dibuat oleh penulis telah selesai dan sudah diuji, baik melalui pengujian *white box* yang dilakukan oleh penulis sendiri tanpa melibatkan *user* maupun pengujian *black box* yang sudah sesuai dengan kebutuhan *user*. Maka sistem tersebut telah siap untuk diterapkan pada objek penelitian.

BAB 4. DESAIN DAN PERANCANGAN SISTEM

Bab ini akan membahas tentang analisis dan perancangan untuk membuat Sistem Informasi Persediaan Barang Berbasis Web Pada CV.Unicorner Banyuwangi Menggunakan Metode *Economic Order Quantity* (EOQ). Tahapan - tahapan perancangan dilaksanakan berdasarkan metode inkremental seperti yang sudah dijelaskan pada bab sebelumnya.

4.1. Studi Pustaka

Mengumpulkan data dari buku-buku referensi, internet, dan jurnal *online* yang berkaitan dengan penelitian tersebut. Salah satunya yaitu jurnal *online* dengan judul Sistem Informasi Penjualan dan Pengontrolan Stok Barang Menggunakan Metode EOQ (*Economic Order Quantity*) Studi Kasus di Bengkel Sukasari Motor.

4.2. Pengumpulan Data

Dalam memperoleh data yang diperlukan dalam penelitian maka penulis menggunakan beberapa metode yang digunakan adalah sebagai berikut:

a. Metode Observasi

Melakukan tinjauan langsung pada perusahaan untuk menganalisa apa saja yang perlu diteliti.

b. Metode Wawancara

Wawancara langsung dengan pihak yang bersangkutan pada CV. Unicorner Banyuwangi untuk mendapatkan data-data yang akurat.

4.3. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengumpulkan data mengenai spesifikasi dari sistem guna memenuhi fungsi yang diperlukan oleh pengguna. Kebutuhan ini meliputi kebutuhan fungsional dan kebutuhan non fungsional.

4.3.1. Kebutuhan Fungsional

Kebutuhan fungsional dari Sistem Informasi Persediaan Barang Berbasis Web Pada CV.Unicorner Banyuwangi Menggunakan Metode *Economic Order Quantity* (EOQ) adalah:

1. Sistem ini dapat memproses data user, data barang, data penjualan dan data pembelian di CV.Unicorner Banyuwangi.
2. Sistem dapat memberikan informasi stok barang yang optimal menggunakan metode *Economic Order Quantity* (EOQ).
3. Sistem dapat menyimpan data-data yang diinputkan kedalam *database*.

4.3.2. Kebutuhan Non Fungsional

Kebutuhan non fungsional dari Sistem Informasi Persediaan Barang Berbasis Web Pada CV.Unicorner Banyuwangi Menggunakan Metode *Economic Order Quantity* (EOQ) adalah:

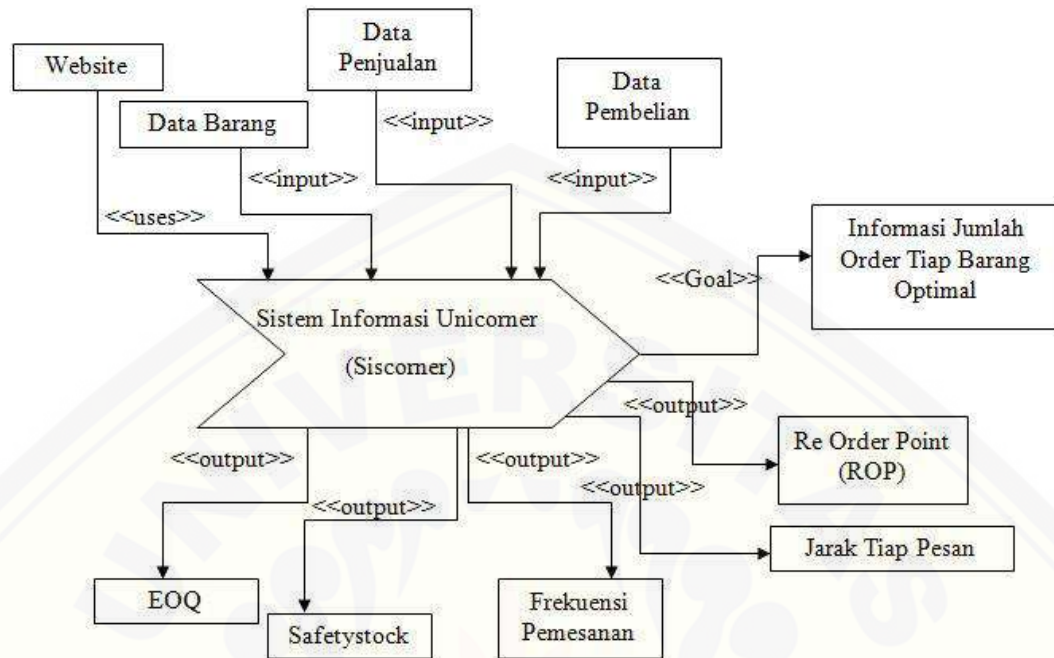
1. Sistem dapat diakses selama 24 jam.
2. Sistem hanya dapat digunakan di PC.
3. Sistem ini dilengkapi dengan password, setiap pengguna memiliki hak aksesnya masing-masing.
4. Tampilan sistem mudah dipahami.
5. Sistem memberi *respon time* yang cepat.

4.4. Desain Sistem

Tahap desain dilakukan setelah mengetahui kebutuhan dan gambaran sistem yang akan dibangun. Sistem yang akan dibangun menggunakan perancangan berbasis *Object-Oriented Programming* (OOP).

4.4.1. Bussines Process

Tahap awal dari pembuatan desain sistem yaitu *bussines process*



Gambar 0.1. Bussines Process

4.4.2. Use case Diagram

Tahap selanjutnya yaitu *Use case Diagram* merupakan gambaran dari interaksi aktor pada fitur-fitur sistem. Pada tahap ini dijelaskan mengenai siapa saja yang memiliki hak akses dan fitur apa saja yang terdapat pada sistem.

(a.) Definisi aktor

Definisi aktor merupakan penjelasan tentang aktor – aktor sebagai pengguna dari sistem informasi persediaan barang di CV. Unicorner. Dari hasil analisis, ada 3 aktor yang nantinya menggunakan sistem ini. Seperti yang dijelaskan pada tabel 4.1 berikut.

Tabel 4.1. Definisi Aktor

No	Aktor	Penjelasan
1	Owner	Owner tidak bisa mengelola fitur-fitur yang ada pada sistem tersebut, hanya dapat login dan melihat data

		penjualan serta pembelian.
2	Admin Gudang	Admin gudang hampir memiliki hak akses penuh terhadap fitur-fitur sistem ini. Fitur-fitur yang dapat dikelola admin gudang yaitu <i>login</i> , <i>user</i> (menambah, mengedit dan menghapus), mengelola data barang (menambah, mengedit dan menghapus), mengelola data pembelian barang (menambah, mengedit dan menghapus), dan manajemen data EOQ.
3	Admin Penjualan	Admin penjualan memiliki hak akses <i>login</i> , dan mengelola data penjualan barang.

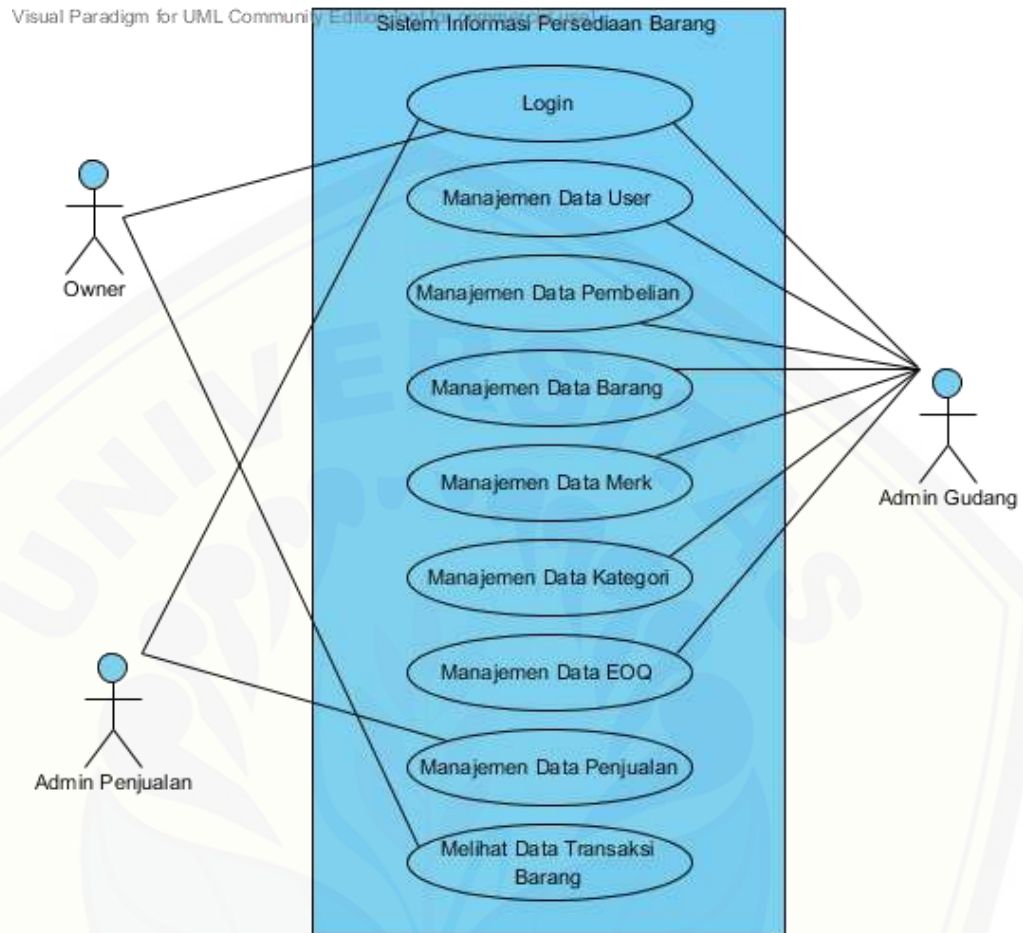
(b.) **Definisi Use case**

Definisi *Use case* merupakan penjelasan dari masing – masing *Use case* atau fitur – fitur pada sistem informasi persediaan barang di CV. Unicorner. Penjelasan *usecase* dapat di lihat pada Tabel 4.2.

Tabel 4.2. Definisi *Use case*

No	<i>Use case</i>	Penjelasan
1.	Login	Merupakan <i>Use case</i> yang menggambarkan proses login atau autentifikasi untuk masuk sistem.
2.	Manajemen data User	Merupakan <i>Use case</i> yang menggambarkan proses manajemen data user yaitu: <ol style="list-style-type: none"> a. Melihat data user b. Menambah data user c. Mengedit data user d. Menghapus data user Proses ini dilakukan oleh admin gudang.
3.	Manajemen data Pembelian	Merupakan <i>Use case</i> yang menggambarkan proses manajemen data pembelian yaitu: <ol style="list-style-type: none"> a. Melihat data pembelian b. Menambah data pembelian c. Mengedit data pembelian d. Menghapus data pembelian Proses ini dilakukan oleh admin gudang.
4.	Manajemen data Barang	
5.	Manajemen data Merk	Merupakan <i>Use case</i> yang menggambarkan proses manajemen data merk yaitu: <ol style="list-style-type: none"> a. Melihat data merk b. Menambah data merk c. Mengedit data merk

	d. Menghapus data merk Proses ini dilakukan oleh admin gudang.
6. Manajemen data Kategori	Merupakan <i>Use case</i> yang menggambarkan proses manajemen data kategori yaitu: a. Melihat data kategori b. Menambah data kategori c. Mengedit data kategori d. Menghapus data kategori Proses ini dilakukan oleh admin gudang.
7. Manajemen data EOQ	
8. Manajemen data Penjualan	Merupakan <i>Use case</i> yang menggambarkan proses manajemen data penjualan yaitu: a. Melihat data penjualan b. Menambah data penjualan c. Mengedit data penjualan d. Menghapus data penjualan Proses ini dilakukan oleh admin penjualan.
9. Melihat data Barang	Merupakan <i>Use case</i> yang menggambarkan proses melihat data barang oleh owner.



Gambar 0.2. Use case Diagram Sistem Informasi Persediaan Barang

4.4.3. Use case Skenario

Use case skenario merupakan penjelasan alur kerja dari fitur-fitur yang ada di dalam *use case* diagram. dokumentasi terhadap kebutuhan fungsional sistem. *Use case* skenario dari sistem informasi persediaan barang adalah sebagai berikut.

(a.) Use case Skenario Login

Use case skenario ini menjelaskan tentang alur dari proses *login* yang dilakukan oleh *user* sesuai dengan hak aksesnya. *Use case* skenario *login* dapat dilihat pada

Lampiran A.1.

(b.) Use case skenario Manajemen Data User

Use case skenario ini menjelaskan tentang alur dari proses manajemen data user yang dilakukan oleh Admin Gudang. *Use case* skenario manajemen data *user* dapat dilihat pada Lampiran A.2.

(c.) Use case skenario Manajemen Data Pembelian

Use case skenario ini menjelaskan tentang alur dari proses manajemen data pembelian yang dilakukan oleh Admin Gudang. *Use case* skenario manajemen data pembelian dapat dilihat pada Lampiran A.3.

(d.) Use case skenario Manajemen Data Barang

Use case skenario ini menjelaskan tentang alur dari proses manajemen data kategori yang dilakukan oleh Admin Gudang. *Use case* skenario manajemen data kategori dapat dilihat pada Lampiran A.4.

(e.) Use case skenario Manajemen Data Merk

Use case skenario ini menjelaskan tentang alur dari proses manajemen data merk yang dilakukan oleh Admin Gudang. *Use case* skenario manajemen data merk dapat dilihat pada Lampiran A.5.

(f.) Use case skenario Manajemen Data Kategori

Use case skenario ini menjelaskan tentang alur dari proses manajemen data kategori yang dilakukan oleh Admin Gudang. *Use case* skenario manajemen data kategori dapat dilihat pada Lampiran A.6.

(g.) Use case skenario Manajemen Data EOQ

Use case skenario ini menjelaskan tentang alur dari proses manajemen data

EOQ yang dilakukan oleh Admin Gudang. *Use case* skenario manajemen data EOQ dapat dilihat pada Tabel 4.3.

Tabel 4.3. *Use case* Skenario Manajemen Data EOQ

Nama		Manajemen Data EOQ	
Aktor		Admin Gudang	
<i>Pre-Condition</i>		Menampilkan data EOQ	
<i>Post-Condition</i>		Admin Gudang berhasil mengelola data EOQ	
SKENARIO NORMAL			
Aktor		Sistem	
1	Memilih Menu “Manajemen Data EOQ”		
		2	Menampilkan form Manajemen Data EOQ
3	Mengisi form data EOQ		
4	Menekan tombol “Hitung”		
		5	Melakukan validasi data
		6	Menampilkan hasil penghitungan EOQ
SKENARIO NORMAL			
Jika aktor ingin menghapus hasil data EOQ			
Aktor		Sistem	
7	Menekan tombol “Hapus”		
		8	Menghapus hasil EOQ
SKENARIO ALTERNATIVE			
Jika aktor mengisi data tidak lengkap			
Aktor		Sistem	
3	Mengisi form data EOQ		
4	Menekan tombol “Hitung”		
		5	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi

(h.) *Use case* skenario Manajemen Penjualan

Use case skenario ini menjelaskan tentang alur dari proses manajemen data penjualan yang dilakukan oleh Admin Penjualan. *Use case* skenario manajemen data

penjualan dapat dilihat pada Lampiran A.7.

(i.) Use case skenario melihat Data Transaksi Barang

Use case skenario melihat data transaksi barang ini menjelaskan tentang alur *Owner* dalam mengakses sistem. Dalam fitur ini *Owner* dapat melihat semua transaksi penjualan dan pembelian barang, namun tidak mempunyai hak untuk mengelolanya. *Use case* skenario melihat data transaksi barang dapat dilihat pada Lampiran A.8.

4.4.4. Activity Diagram

Activity Diagram merupakan urutan aktivitas pada sistem untuk membantu memahami proses yang terjadi didalam suatu fungsi fitur sistem. Sistem Informasi Persediaan Barang ini memiliki 9 *Activity* Diagram yaitu sebagai berikut

(a.) Activity Diagram Login

Activity diagram alur aktivitas dari proses *Login* dijelaskan secara lengkap pada Lampiran B.1.

(b.) Activity Diagram Manajemen Data User

Activity diagram alur aktivitas dari proses manajemen data *user* dijelaskan secara lengkap pada Lampiran B.2.

(c.) Activity Diagram Manajemen Data Pembelian

Activity diagram alur aktivitas dari proses manajemen data pembelian dijelaskan secara lengkap pada Lampiran B.3.

(d.) Activity Diagram Manajemen Data Barang

Activity diagram alur aktivitas dari proses manajemen data barang dijelaskan

secara lengkap pada Lampiran B.4.

(e.) Activity Diagram Manajemen Data Merk

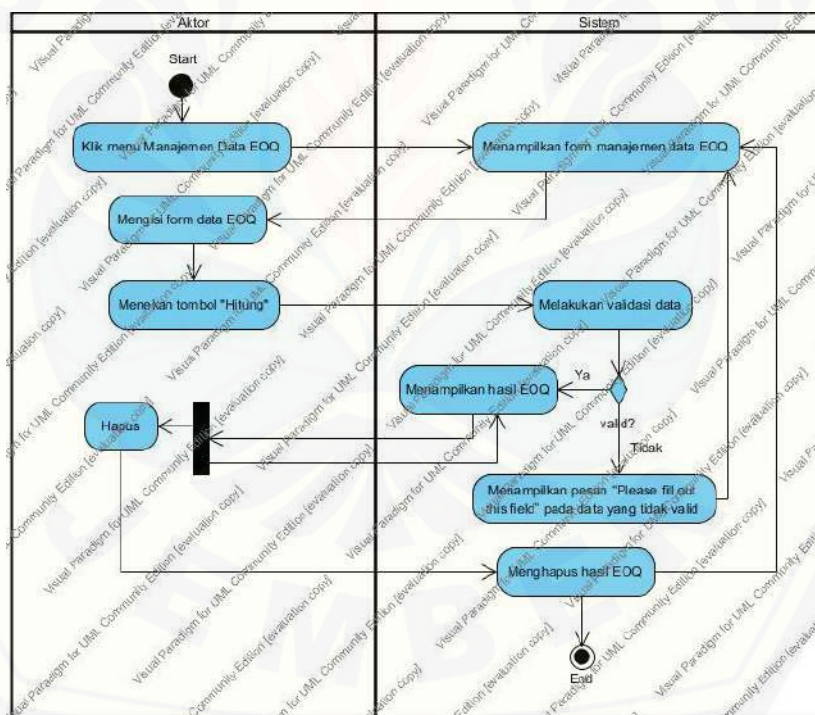
Activity diagram alur aktivitas dari proses manajemen data merk dijelaskan secara lengkap pada Lampiran B.5.

(f.) Activity Diagram Manajemen Data Kategori

Activity diagram alur aktivitas dari proses manajemen data kategori dijelaskan secara lengkap pada Lampiran B.6.

(g.) Activity Diagram Manajemen Data EOQ

Activity diagram alur aktivitas dari proses manajemen data EOQ dijelaskan secara lengkap pada Gambar 4.3.



Gambar 0.3. Activity Diagram Manajemen Data EOQ

(h.) Activity Diagram Manajemen Data Penjualan

Activity diagram alur aktivitas dari proses manajemen data penjualan dijelaskan secara lengkap pada Lampiran B.7.

(i.) ***Activity Diagram Melihat Data Transaksi Barang***

Activity diagram alur aktivitas dari proses manajemen data transaksi barang dijelaskan secara lengkap pada Lampiran B.8.

4.4.5. Sequence Diagram

Sequence diagram atau diagram urutan adalah suatu diagram yang menampilkan urutan interaksi antar objek di dalam sistem. *Sequence* diagram digunakan untuk menggambarkan perilaku objek terhadap *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirim dan diterima antar objek. *Sequence* diagram dari Sistem Informasi Persediaan Barang adalah sebagai berikut.

(a.) ***Sequence Diagram Login***

Alur aktivitas *Sequence* diagram dalam fitur *login* dijelaskan secara lengkap pada Lampiran C.1.

(b.) ***Sequence Diagram Manajemen Data User***

Alur aktivitas *Sequence* diagram dalam fitur manajemen data *user* dijelaskan secara lengkap pada Lampiran C.2.

(c.) ***Sequence Diagram Manajemen Data Pembelian***

Alur aktivitas *Sequence* diagram dalam fitur manajemen data pembelian dijelaskan secara lengkap pada Lampiran C.3.

(d.) ***Sequence Diagram Manajemen Data Barang***

Alur aktivitas *Sequence* diagram dalam fitur manajemen data barang dijelaskan secara lengkap pada Lampiran C.4.

(e.) *Sequence Diagram Manajemen Data Merk*

Alur aktivitas *Sequence* diagram dalam fitur manajemen data merk dijelaskan secara lengkap pada Lampiran C.5.

(f.) *Sequence Diagram Manajemen Data Kategori*

Alur aktivitas *Sequence* diagram dalam fitur manajemen data kategori dijelaskan secara lengkap pada Lampiran C.6

(g.) *Sequence Diagram Manajemen Data EOQ*

Alur aktivitas *Sequence* diagram dalam fitur manajemen data EOQ dijelaskan secara lengkap pada gambar 4.4.

Gambar 0.4. *Sequence* Manajemen Data EOQ

(h.) *Sequence Diagram Manajemen Data Penjualan*

Alur aktivitas *Sequence* diagram dalam fitur manajemen data penjualan dijelaskan secara lengkap pada Lampiran C.7.

(i.) *Sequence Diagram Melihat Data Transaksi Barang*

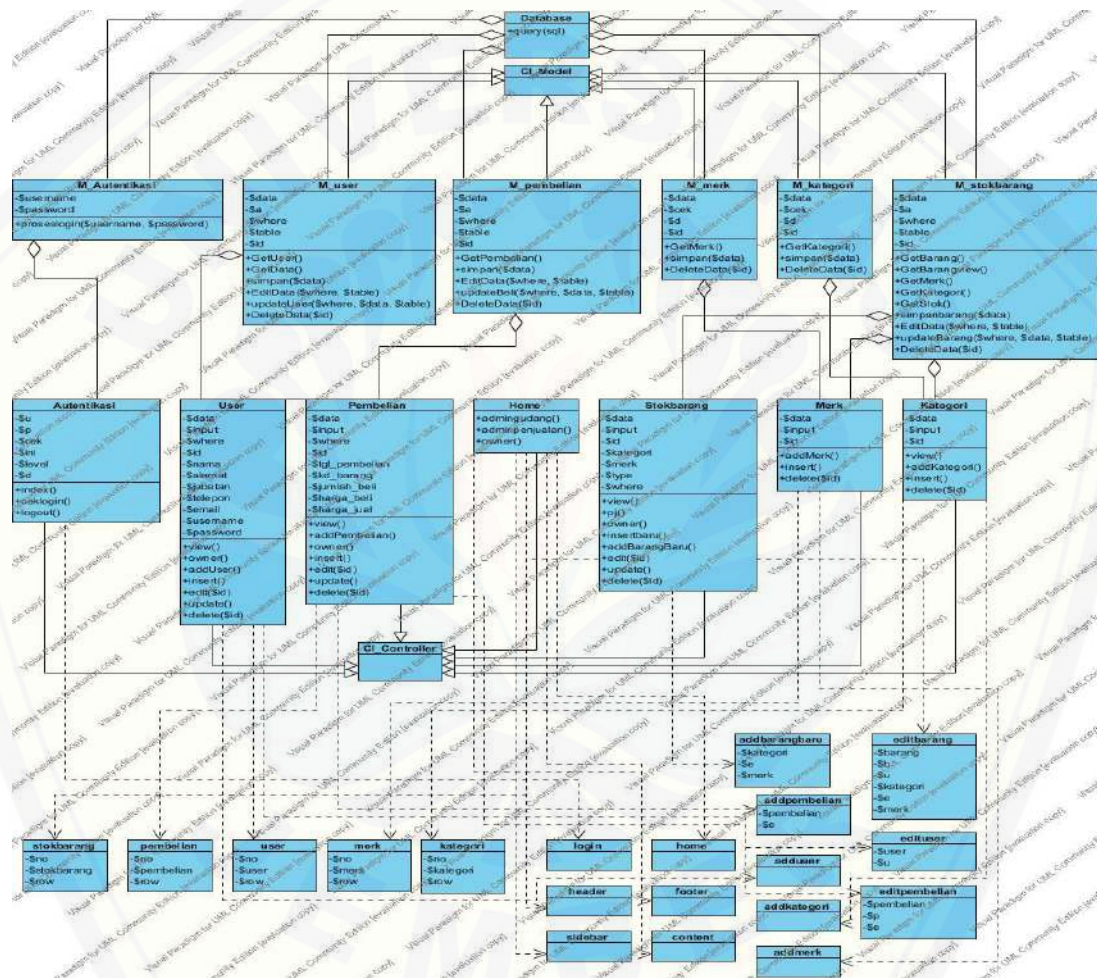
Alur aktivitas *Sequence* diagram dalam fitur melihat data transaksi barang dijelaskan secara lengkap pada Lampiran C.8.

4.4.6. *Class Diagram*

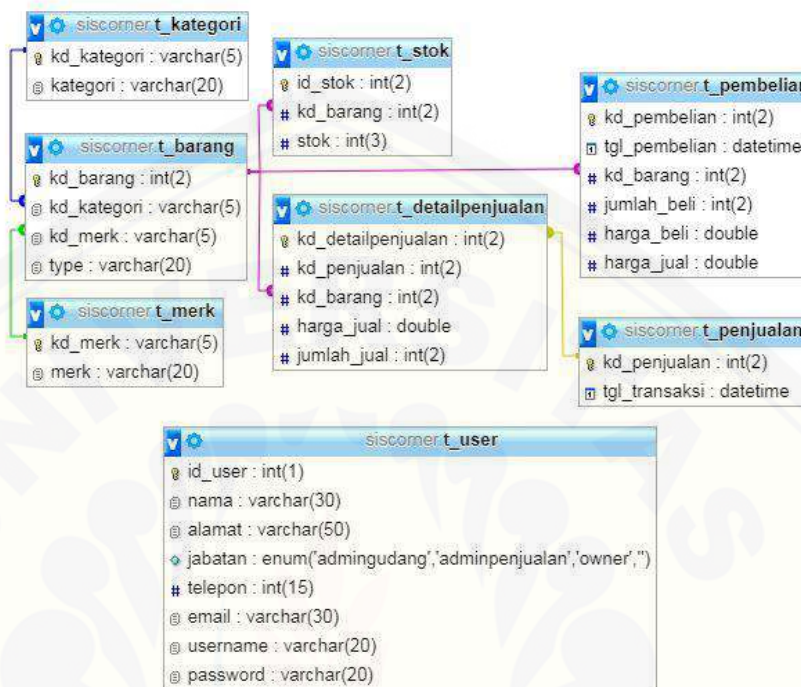
Class diagram menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan. *Class* diagram Sistem Informasi Persediaan Barang dapat dilihat pada gambar 4.5.

4.4.7. Entity Relationship Diagrams (ERD)

Entity Relationship Diagrams (ERD) adalah dokumentasi model untuk menjelaskan hubungan antar data dalam basis data berdasarkan Entity yang mempunyai hubungan antar relasi. ERD Sistem Informasi Persediaan Barang dapat dilihat pada gambar 4.6.



Gambar 0.5. Class Diagram Sistem Informasi Persediaan Barang



Gambar 4.6. ERD Sistem Informasi Persediaan Barang

4.5. Penulisan Kode Program

Penulisan kode program merupakan tahap lanjutan dari desain sistem, desain sistem yang telah dibuat berdasarkan pemodelan UML diimplementasikan kedalam kode program. Penulisan kode program menggunakan bahasa *Page Hypertext Pre-Processor* (PHP) dengan framework *Code Igniter*. Penulisan kode program dibuat berdasarkan fitur-fitur yang ada pada *use case*. Penulisan kode program bisa dilihat pada lampiran D.

4.6. Pengujian Program

Pengujian ini dilakukan dengan dua metode pengujian yaitu *White Box Testing* dan *Black Box Testing*.

4.6.1. Pengujian White box

Pengujian *white box* merupakan pengujian dari kode - kode program yang meliputi *listing* program, grafik alir, kompleksitas siklomatis, *basisset* dan *test case*.

(a.) Manajemen EOQ

Pengujian *white box* pada fitur manajemen EOQ adalah sebagai berikut:

1. Listing Manajemen EOQ

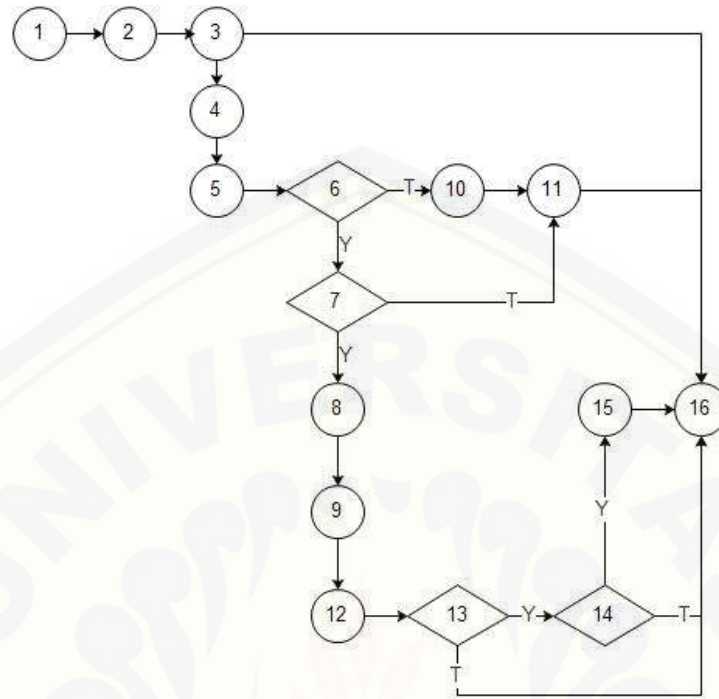
```

1 public function view(){
2     $this->load->view('admingudang/header');
3     $this->load->view('admingudang/eoq');
4     $this->load->view('footer');
5 }
6 public function insert(){
7     $awal=1;$seq=0;
8     if (@$_POST["barang"]!=''){
9         if (empty($_SESSION["isi"])==TRUE){
10            $_SESSION["isi"]=1;
11        }else{
12            $_SESSION["isi"]++;
13        }
14        @$barang=trim($_POST["barang"]);
15        @$max=trim($_POST["max"]);
16        @$min=trim($_POST["min"]);
17        @$seq=sqrt($barang*12000/5000);
18        @$frekuensi=$barang/$seq;
19        @$jarak=365/$frekuensi;
20        @$ss=( $max-$min)*3;
21        @$rop=( $barang/12)*3+$ss;
22        $_SESSION["akhir"][$_SESSION["isi"]]=array($barang,$seq,$frekuensi,$jarak,$ss,$rop);
23    }else{
24        echo "<h3 style='color:red;'>isi dulu barang anda!</h3>";
25    }
26    @$awal = $_SESSION["isi"];
27    for ($i=0;$i<=$awal;$i++){
28        if (@$_SESSION['akhir'][$i][0]!=''){
29            echo "<tr>
30                <td>$i</td>
31                <td>".$_SESSION['akhir'][$i][0]."</td>
32                <td align=right>".$_SESSION['akhir'][$i][1]."</td>
33                <td align=right>".$_SESSION['akhir'][$i][2]."</td>
34                <td align=right>".$_SESSION['akhir'][$i][3]."</td>
35                <td align=right>".$_SESSION['akhir'][$i][4]."</td>
36                <td align=right>".$_SESSION['akhir'][$i][5]."</td>
37                <td align=right>".$_SESSION['akhir'][$i][6]."</td></tr>";
38        }
39        $seq=$_SESSION['akhir'][$i][6]+$seq;
40    }

```

2. Grafik Alir

Grafik Alir fitur Manajemen EOQ. Grafik alir tersebut di gambarkan dengan simpel dan garis dapat di lihat pada Gambar 4.7.



Gambar 4.7. Diagram Alir Manajemen EOQ

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur manajemen EOQ dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 20 - 16 + 2 = 6$$

4. Basis Set

Basis set dari fitur manajemen EOQ menghasilkan jalur yaitu :

Jalur 1 = 1-2-3-16

Jalur 2 = 1-2-3-4-5-6-10-11-16

Jalur 3 = 1-2-3-4-5-6-7-11-16

Jalur 4 = 1-2-3-4-5-6-7-8-9-12-13-14-15-16

Jalur 5 = 1-2-3-4-5-6-7-8-9-12-13-14-16

Jalur 6 = 1-2-3-4-5-6-7-8-9-12-13-16

5. Test Case

Tabel 4.4. Manajemen EOQ Jalur 1

Test case	<i>User</i> menekan menu EOQ
Target yang diharapkan	Menampilkan halaman manajemen EOQ
Hasil pengujian	Benar
Jalur	Jalur 1
File	Eoq.php

Tabel 4.5. Manajemen EOQ Jalur 2

Test case	<i>User</i> menekan tombol “Hitung” tanpa mengisi total barang terjual pada form EOQ
Target yang diharapkan	Menampilkan halaman EOQ yang kosong
Hasil pengujian	Benar
Jalur	Jalur 2
File	Eoq.php

Tabel 4.6. Manajemen EOQ Jalur 3

Test case	<i>User</i> mengisi form penghitungan EOQ tidak lengkap kemudian menekan tombol Hitung
Target yang diharapkan	Menampilkan penghitungan EOQ tidak lengkap
Hasil pengujian	Benar
Jalur	Jalur 3
File	Eoq.php

Tabel 4.7. Manajemen EOQ Jalur 4

Test case	<i>User</i> mengisi form penghitungan EOQ secara lengkap kemudian menekan tombol Hitung
Target yang diharapkan	Menampilkan hasil penghitungan EOQ lengkap
Hasil pengujian	Benar
Jalur	Jalur 4
File	Eoq.php

Tabel 4.8. Manajemen EOQ Jalur 5

Test case	<i>User</i> mengisi kembali form penghitungan EOQ tanpa menghapus penghitungan sebelumnya kemudian menekan tombol Hitung
Target yang diharapkan	Menampilkan hasil penghitungan EOQ yang sebelumnya dan yang baru
Hasil pengujian	Benar
Jalur	Jalur 5
File	Eoq.php

Tabel 4.9. Manajemen EOQ Jalur 6

Test case	<i>User</i> mengisi kembali form penghitungan EOQ
-----------	---

	tanpa menghapus penghitungan sebelumnya kemudian menekan tombol Hitung
Target yang diharapkan	Menampilkan hasil penghitungan EOQ yang sebelumnya dan yang baru
Hasil pengujian	Benar
Jalur	Jalur 6
File	Eoq.php

4.6.2. Pengujian *Black box*

Pengujian *Black box* dilakukan untuk mengetahui apakah *input* dan *output* dari sistem sesuai dengan kebutuhan fungsional atau tidak. Pengujian dilakukan pada *form* untuk setiap *usecase*. Pengujian ini dilakukan oleh calon pengguna Sistem Informasi Persediaan Barang Berbasis Web pada CV.Unicorner Banyuwangi Menggunakan Metode *Economic Order Quantity*.

4.7. Pemeliharaan

Tahap ini merupakan tahap terakhir dari pembuatan Sistem Informasi Persediaan Barang. Pemeliharaan dilakukan setelah sistem selesai dibuat dan telah dilakukan pengujian. Seiring dengan penggunaan sistem secara terus menerus oleh user, maka akan memungkinkan jika sistem ini terdapat bugs. Jadi tahapan ini harus terus dilakukan untuk memperbaiki bugs dan kemungkinan adanya perbaikan fitur.

BAB 6. PENUTUP

Bab ini merupakan bagian akhir di dalam penulisan skripsi, berisi tentang kesimpulan dan saran. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Sistem Informasi Persediaan Barang memiliki beberapa fitur yaitu, Login, Manajemen data user, Manajemen data pembelian, Manajemen data barang, Manajemen data merk, Manajemen data kategori, Manajemen data EOQ, Manajemen data penjualan dan Melihat data transaksi barang.
2. Sistem Informasi Persediaan Barang menggunakan metode Economic Order Quantity ini masih harus dilakukan banyak perbaikan, karena masih belum bekerja secara maksimal.

6.2 Saran

Sistem ini masih belum sempurna, hanya terdapat fitur dasar untuk memprediksi stok barang untuk periode selanjutnya. Itupun masih belum secara otomatis karena masih dengan cara menginputkan data satu persatu. Untuk kedepannya diharapkan bisa lebih disempurnakan dan diberi fitur tambahan seperti print laporan dan nota penjualan, serta laporan keuangan untuk penghitungan laba dan rugi.

DAFTAR PUSTAKA

- Abidin, Faqih. (2011). *Sistem Informasi Pengontrolan Stok Barang Menggunakan Metode EOQ (Economic Order Quantity) Studi Kasus Bengkel Sukasari Motor*.
- Assauri, S. (2004). *Manajemen Produksi dan Operasi*. Jakarta: CP-FEUI.
- Ginting, I. (2013). Perancangan Sistem Pengontrolan Stok Barang pada Blesscom Komputer dengan Metode Economic Order Quantity (EOQ). *Pelita Informatika Budi Darma*, IV, 4.
- Rangkuti, F. (2004). *Manajemen Persediaan* (2 ed.). Jakarta: Rajawali Pers.
- Ristono, A. (2009). *Manajemen Persediaan*. Yogyakarta: Graha Ilmu.
- Riyanto, B. (2001). *Dasar-Dasar Pembelajaran Perusahaan* (4 ed.). Yogyakarta: BPFE.
- S. Rosa A dan Salahuddin, M. (2014). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek* (2 ed.). Bandung: Informatika.
- Sartono, A. (2010). *Manajemen Keuangan Teori dan Aplikasi* (4 ed.). Yogyakarta: BPFE.

LAMPIRAN

LAMPIRAN A. USECASE SKENARIO

Lampiran A.1. Use case Skenario Autentikasi (*Login*)

Nama		Autentikasi (<i>Login</i>)	
Aktor		Admin Gudang, Admin Penjualan dan <i>Owner</i>	
<i>Pre-Condition</i>		Admin gudang, Admin penjualan dan <i>Owner</i> harus memiliki username dan password dengan hak akses masing-masing.	
<i>Post-Condition</i>		Admin gudang, Admin penjualan dan <i>Owner</i> berhasil masuk sistem.	
SKENARIO NORMAL			
Aktor		Sistem	
1	Menjalankan sistem		
		2	Menampilkan form <i>login</i>
3	Mengisi <i>username</i> dan <i>password</i>		
4	Menekan tombol " <i>login</i> "		
		5	Melakukan validasi <i>username</i> dan <i>password</i>
		6	Menampilkan halaman sesuai hak akses
SKENARIO ALTERNATIF			
Jika aktor tidak mengisi form <i>login</i> secara lengkap			
Aktor		Sistem	
4	Menekan tombol <i>login</i> tetapi data tidak diisi dengan lengkap		
		5	Menampilkan form <i>login</i>
SKENARIO ALTERNATIF			
Jika aktor salah memasukan <i>username</i> atau <i>password</i>			
Aktor		Sistem	
4	Menekan tombol <i>login</i> tetapi <i>password</i> atau <i>username</i> salah		
		5	Melakukan validasi <i>username</i> dan <i>password</i>
		6	Menampilkan form <i>login</i>

Lampiran A.2. Use case Skenario Manajemen Data User

Nama		Manajemen Data User	
Aktor		Admin Gudang	
Pre-Condition		Menampilkan data user	
Post-Condition		Admin Gudang berhasil mengelola data user	
SKENARIO NORMAL			
Aktor		Sistem	
1	Memilih Menu "user"		
		2	Menampilkan halaman data user
SKENARIO NORMAL			
Jika aktor ingin menambah user			
Aktor		Sistem	
3	Menekan tombol "Tambah Data"		
		4	Menampilkan form tambah user
5	Mengisi form tambah user		
6	Menekan tombol "Simpan"		
		7	Melakukan validasi data
		8	Menyimpan data kedalam database
		9	Menampilkan data user
SKENARIO ALTERNATIF			
Jika aktor menambah user tetapi data yang diinputkan tidak lengkap			
Aktor		Sistem	
5	Mengisi form tambah user tetapi data tidak lengkap		
6	Menekan tombol "Simpan"		
		7	Menampilkan pesan "Please fill out this field" pada form yang belum terisi
SKENARIO NORMAL			
Jika aktor ingin menghapus data user			
Aktor		Sistem	
10	Menekan tombol "Hapus" pada kolom action di tabel data user sesuai nama user yang ingin dihapus		

		11	Menghapus data yang dipilih dan menampilkan halaman data <i>user</i>
SKENARIO NORMAL			
Jika aktor ingin mengedit data <i>user</i>			
Aktor		Sistem	
10	Menekan tombol “Edit” pada kolom <i>action</i> di tabel data user sesuai nama <i>user</i> yang ingin diedit		
		11	Menampilkan halaman edit data <i>user</i> yang sudah berisi data <i>user</i> yang dipilih
12	Melakukan edit data		
13	Menekan tombol “Simpan”		
		14	Melakukan validasi data
		15	Melakukan update <i>database</i>
		16	Menampilkan halaman data <i>user</i> yang telah terupdate
SKENARIO ALTERNATIF			
Jika aktor ingin mengedit data user tetapi data tidak lengkap			
Aktor		Sistem	
12	Melakukan edit data tetapi data yang dimasukan tidak lengkap		
13	Menekan tombol “Simpan”		
		14	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi
		13	Menampilkan halaman data <i>user</i>

Lampian A.3. Use case Skenario Manajemen Data Pembelian

Nama	Manajemen Data Pembelian
Aktor	Admin Gudang
<i>Pre-Condition</i>	Menampilkan data pembelian
<i>Post-Condition</i>	Admin Gudang berhasil mengelola data pembelian
SKENARIO NORMAL	
Aktor	
Sistem	
1	Memilih Menu “Pembelian”

		2	Menampilkan halaman data pembelian
SKENARIO NORMAL			
Jika aktor ingin menambah pembelian			
Aktor		Sistem	
3	Menekan tombol “Tambah Data”		
		4	Menampilkan form tambah pembelian
5	Mengisi form tambah pembelian		
6	Menekan tombol “Simpan”		
		7	Melakukan validasi data
		8	Menyimpan data kedalam <i>database</i>
		9	Menampilkan data pembelian
SKENARIO ALTERNATIF			
Jika aktor menambah pembelian tetapi data yang diinputkan tidak lengkap			
Aktor		Sistem	
5	Mengisi form tambah pembelian tetapi data tidak lengkap		
6	Menekan tombol “Simpan”		
		7	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi
SKENARIO NORMAL			
Jika aktor ingin menghapus data pembelian			
Aktor		Sistem	
10	Menekan tombol “Hapus” pada kolom <i>action</i> di tabel data pembelian sesuai nama pembelian yang ingin dihapus		
		11	Menghapus data yang dipilih dan menampilkan halaman data pembelian
SKENARIO NORMAL			
Jika aktor ingin mengedit data pembelian			
Aktor		Sistem	
10	Menekan tombol “Edit” pada kolom <i>action</i> di tabel data		

	pembelian sesuai nama pembelian yang ingin diedit		
		11	Menampilkan halaman edit data pembelian yang sudah berisi data pembelian yang dipilih
12	Melakukan edit data		
13	Menekan tombol “Simpan”		
		14	Melakukan validasi data
		15	Melakukan update <i>database</i>
		16	Menampilkan halaman data pembelian yang telah terupdate
SKENARIO ALTERNATIF			
Jika aktor ingin mengedit data pembelian tetapi data tidak lengkap			
Aktor		Sistem	
12	Melakukan edit data tetapi data yang dimasukan tidak lengkap		
13	Menekan tombol “Simpan”		
		14	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi

Lampiran A4. Use case Skenario Manajemen Data Barang

Nama	Manajemen Data Barang		
Aktor	Admin Gudang		
<i>Pre-Condition</i>	Manajemen Data Barang		
<i>Post-Condition</i>	Admin Gudang berhasil mengelola data barang		
SKENARIO NORMAL			
Aktor		Sistem	
1	Memilih Menu “Data Barang”		
		2	Menampilkan halaman data barang
SKENARIO NORMAL			
Jika aktor ingin menambah barang			
Aktor		Sistem	
3	Menekan tombol “Tambah Barang”		
		4	Menampilkan form tambah barang
5	Mengisi form tambah barang		

6	Menekan tombol “Simpan”		
		7	Melakukan validasi data
		8	Menyimpan data kedalam <i>database</i>
		9	Menampilkan data barang
SKENARIO ALTERNATIF			
Jika aktor menambah barang tetapi data yang diinputkan tidak lengkap			
Aktor		Sistem	
5	Mengisi form tambah barang tetapi data tidak lengkap		
6	Menekan tombol “Simpan”		
		7	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi
SKENARIO NORMAL			
Jika aktor ingin menghapus data barang			
Aktor		Sistem	
10	Menekan tombol “Hapus” pada kolom <i>action</i> di tabel data barang sesuai nama barang yang ingin dihapus		
		11	Menghapus data yang dipilih dan menampilkan halaman data barang
SKENARIO NORMAL			
Jika aktor ingin mengedit data barang			
Aktor		Sistem	
10	Menekan tombol “Edit” pada kolom <i>action</i> di tabel data barang sesuai nama barang yang ingin diedit		
		11	Menampilkan halaman edit data barang yang sudah berisi data barang yang dipilih
12	Melakukan edit data barang		
13	Menekan tombol “Simpan”		
		14	Melakukan validasi data
		15	Melakukan update <i>database</i>
		16	Menampilkan halaman data barang yang telah terupdate

SKENARIO ALTERNATIF			
Jika aktor ingin mengedit data barang tetapi data tidak lengkap			
Aktor		Sistem	
12	Melakukan edit data barang tetapi data yang dimasukkan tidak lengkap		
13	Menekan tombol “Simpan”		
		14	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi

Lampiran A.5. Use case Skenario Manajemen Data Merk

Nama	Manajemen Data Merk		
Aktor	Admin Gudang		
<i>Pre-Condition</i>	Menampilkan data merk		
<i>Post-Condition</i>	Admin Gudang berhasil mengelola data merk		
SKENARIO NORMAL			
Aktor		Sistem	
1	Memilih Menu “Merk”		
		2	Menampilkan halaman data merk
SKENARIO NORMAL			
Jika aktor ingin menambah merk			
Aktor		Sistem	
3	Menekan tombol “Tambah Data”		
		4	Menampilkan form tambah merk
5	Mengisi form tambah merk		
6	Menekan tombol “Simpan”		
		7	Melakukan validasi data
		8	Menyimpan data kedalam <i>database</i>
		9	Menampilkan data merk
SKENARIO ALTERNATIF			
Jika aktor menambah merk tetapi data yang diinputkan tidak lengkap			
Aktor		Sistem	
5	Mengisi form tambah merk tetapi data tidak lengkap		
6	Menekan tombol “Simpan”		

		7	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi
SKENARIO NORMAL			
Jika aktor ingin menghapus data merk			
Aktor		Sistem	
10	Menekan tombol “Hapus” pada kolom <i>action</i> di tabel data merk sesuai nama merk yang ingin dihapus		
		11	Menghapus data yang dipilih dan menampilkan halaman data merk
SKENARIO ALTERNATIF			
Jika aktor menginputkan kode merk yang sudah ada			
Aktor		Sistem	
5	Mengisi form merk dengan kode merk yang sudah ada		
6	Menekan tombol “Simpan”		
		7	Menampilkan halaman data merk tetapi data yang diinputkan tidak ada pada tabel

Lampiran A.6. Use case Skenario Manajemen Data Kategori

Nama	Manajemen Data Kategori		
Aktor	Admin Gudang		
<i>Pre-Condition</i>	Menampilkan data kategori		
<i>Post-Condition</i>	Admin Gudang berhasil mengelola data kategori		
SKENARIO NORMAL			
Aktor		Sistem	
1	Memilih Menu “Kategori”		
		2	Menampilkan halaman data kategori
SKENARIO NORMAL			
Jika aktor ingin menambah kategori			
Aktor		Sistem	
3	Menekan tombol “Tambah Data”		

		4	Menampilkan form tambah kategori
5	Mengisi form tambah kategori		
6	Menekan tombol “Simpan”		
		7	Melakukan validasi data
		8	Menyimpan data kedalam <i>database</i>
		9	Menampilkan data kategori
SKENARIO ALTERNATIF			
Jika aktor menambah kategori tetapi data yang diinputkan tidak lengkap			
Aktor		Sistem	
5	Mengisi form tambah kategori tetapi data tidak lengkap		
6	Menekan tombol “Simpan”		
		7	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi
SKENARIO NORMAL			
Jika aktor ingin menghapus data kategori			
Aktor		Sistem	
10	Menekan tombol “Hapus” pada kolom <i>action</i> di tabel data kategori sesuai nama kategori yang ingin dihapus		
		11	Menghapus data yang dipilih dan menampilkan halaman data kategori
SKENARIO ALTERNATIF			
Jika aktor menginputkan kode merk yang sudah ada			
Aktor		Sistem	
5	Mengisi form merk dengan kode kategori yang sudah ada		
6	Menekan tombol “Simpan”		
		7	Menampilkan halaman data kategori tetapi data yang diinputkan tidak ada pada tabel

Lampiran A.7. Use case Skenario Manajemen Data Penjualan

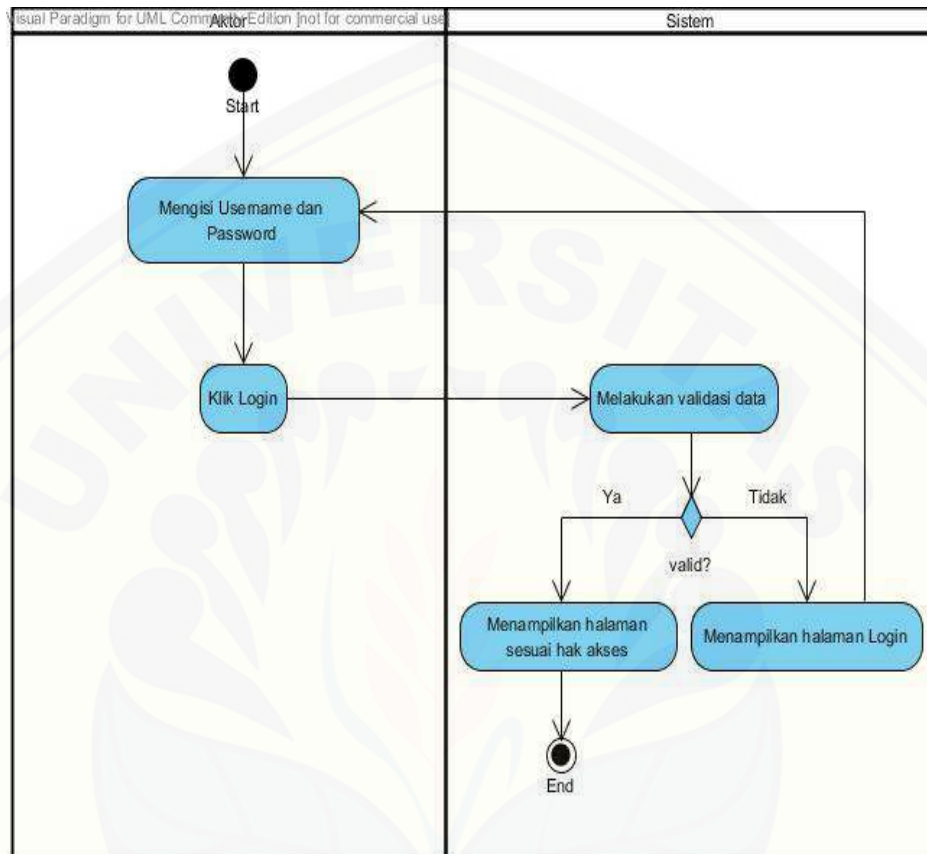
Nama	Manajemen Data Penjualan		
Aktor	Admin Penjualan		
<i>Pre-Condition</i>	Menampilkan data penjualan		
<i>Post-Condition</i>	Admin Penjualan berhasil mengelola data penjualan		
SKENARIO NORMAL			
Aktor		Sistem	
1	Memilih Menu “Penjualan”		
		2	Menampilkan halaman data penjualan
SKENARIO NORMAL			
Jika aktor ingin menambah penjualan			
Aktor		Sistem	
3	Menekan tombol “Tambah Data”		
		4	Menampilkan form tambah penjualan
5	Mengisi form tambah penjualan		
6	Menekan tombol “Simpan”		
		7	Melakukan validasi data
		8	Menyimpan data kedalam <i>database</i>
		9	Menampilkan data penjualan
SKENARIO ALTERNATIF			
Jika aktor menambah penjualan tetapi data yang diinputkan tidak lengkap			
Aktor		Sistem	
5	Mengisi form tambah penjualan tetapi data tidak lengkap		
6	Menekan tombol “Simpan”		
		7	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi
SKENARIO NORMAL			
Jika aktor ingin menghapus data penjualan			
Aktor		Sistem	
10	Menekan tombol “Hapus” pada kolom <i>action</i> di tabel data penjualan sesuai nama penjualan		

	yang ingin dihapus		
		11	Menghapus data yang dipilih dan menampilkan halaman data penjualan
SKENARIO NORMAL			
Jika aktor ingin mengedit data penjualan			
Aktor		Sistem	
10	Menekan tombol “Edit” pada kolom <i>action</i> di tabel data penjualan sesuai nama penjualan yang ingin diedit		
		11	Menampilkan halaman edit data penjualan yang sudah berisi data penjualan yang dipilih
12	Melakukan edit data penjualan		
13	Menekan tombol “Simpan”		
		14	Melakukan validasi data
		15	Melakukan update <i>database</i>
		16	Menampilkan halaman data penjualan yang telah terupdate
SKENARIO ALTERNATIF			
Jika aktor ingin mengedit data penjualan tetapi data tidak lengkap			
Aktor		Sistem	
12	Melakukan edit data penjualan tetapi data yang dimasukkan tidak lengkap		
13	Menekan tombol “Simpan”		
		14	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi

Lampiran A.8. Use case Skenario Melihat Data Transaksi Barang

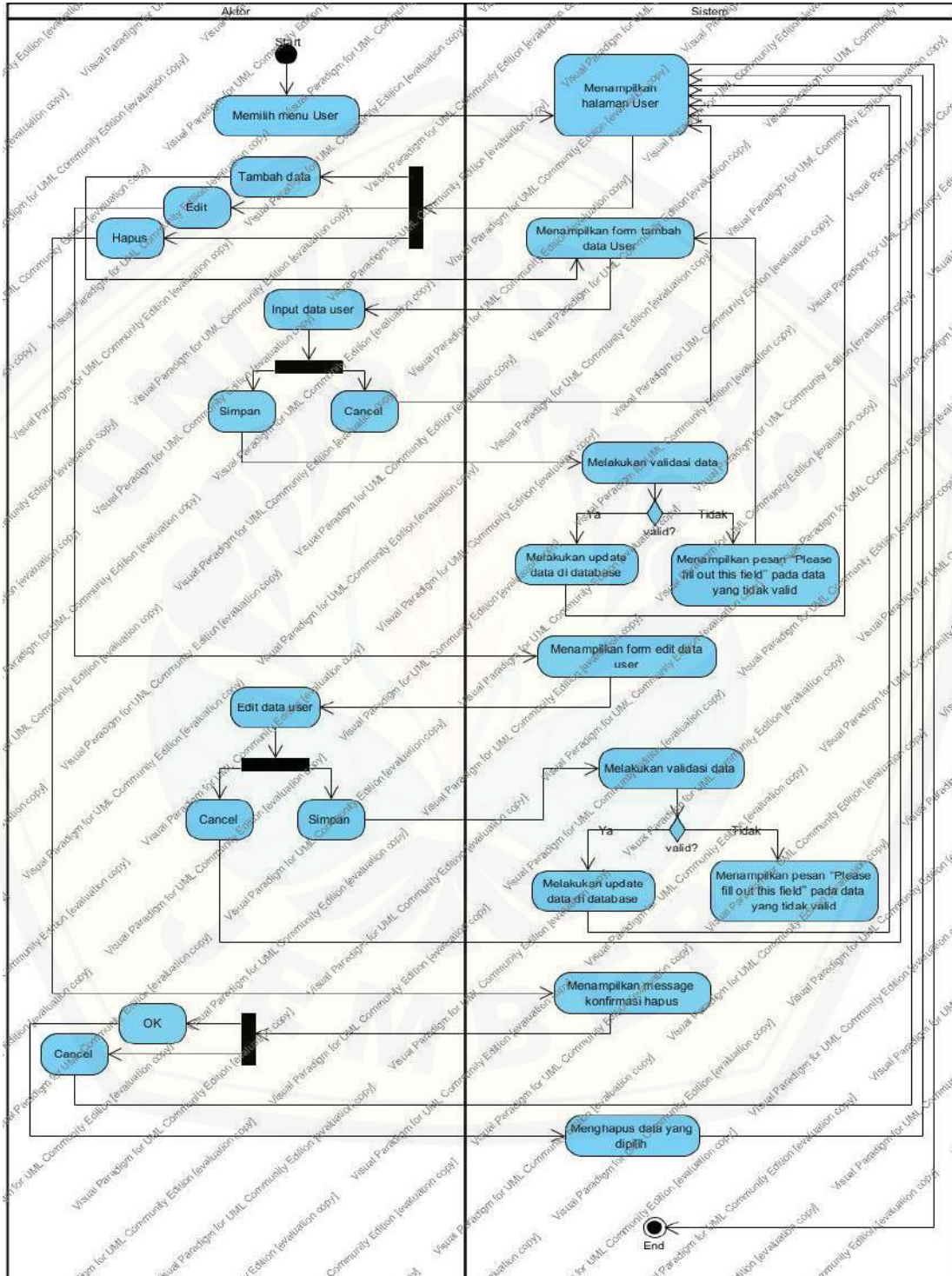
Nama	Melihat Data Transaksi Barang
Aktor	<i>Owner</i>
<i>Pre-Condition</i>	<i>Owner</i> harus login terlebih dahulu
<i>Post-Condition</i>	<i>Owner</i> berhasil melihat data transaksi barang
SKENARIO NORMAL	

Jika <i>Owner</i> ingin melihat data barang			
Aktor		Sistem	
1	Memilih menu “Data Barang”		
		2	Menampilkan detail data barang
SKENARIO NORMAL			
Jika <i>Owner</i> ingin melihat data pembelian			
Aktor		Sistem	
1	Memilih menu “Pembelian”		
		2	Menampilkan detail data pembelian
SKENARIO NORMAL			
Jika <i>Owner</i> ingin melihat data penjualan			
Aktor		Aktor	
1	Memilih menu “Penjualan”		
		2	Menampilkan detail data penjualan

LAMPIRAN B. ACTIVITY DIAGRAM**Lampiran B.1. Activity diagram Autentikasi(login)**

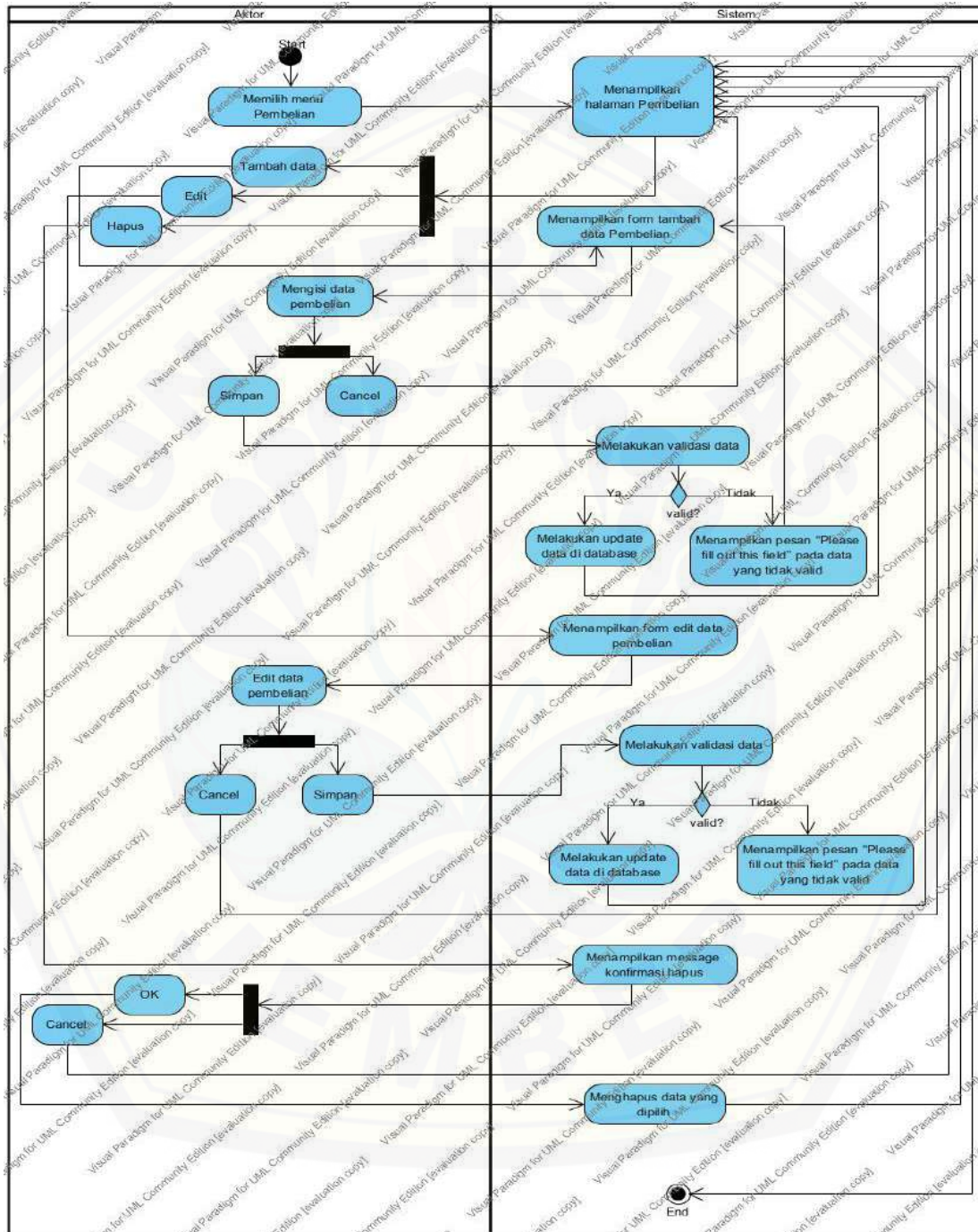
Gambar B.1. Activity diagram login

Lampiran B.2. Activity diagram manajemen data user



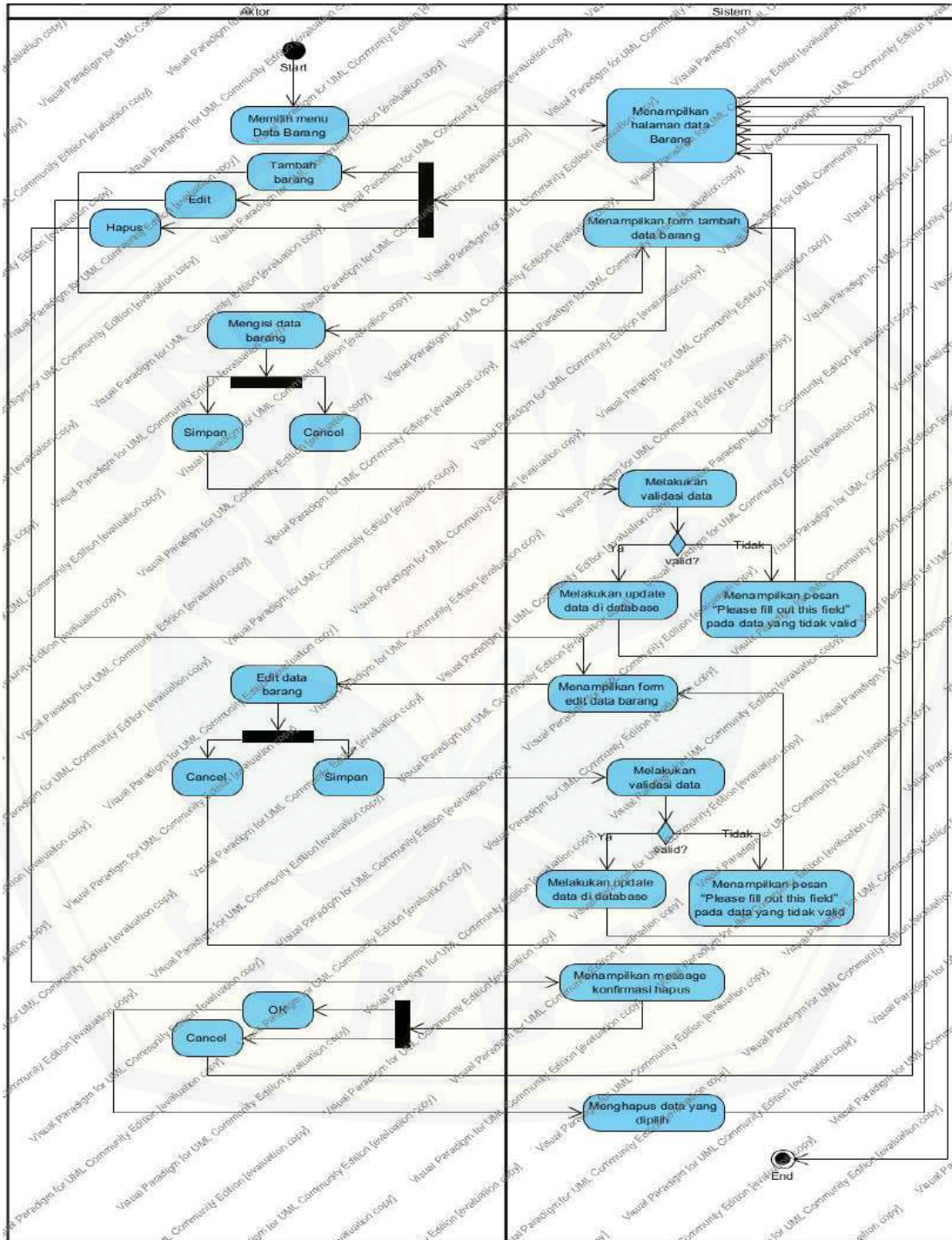
Gambar B.2. Activity diagram manajemen data user

Lampiran B.3. Activity diagram manajemen data pembelian



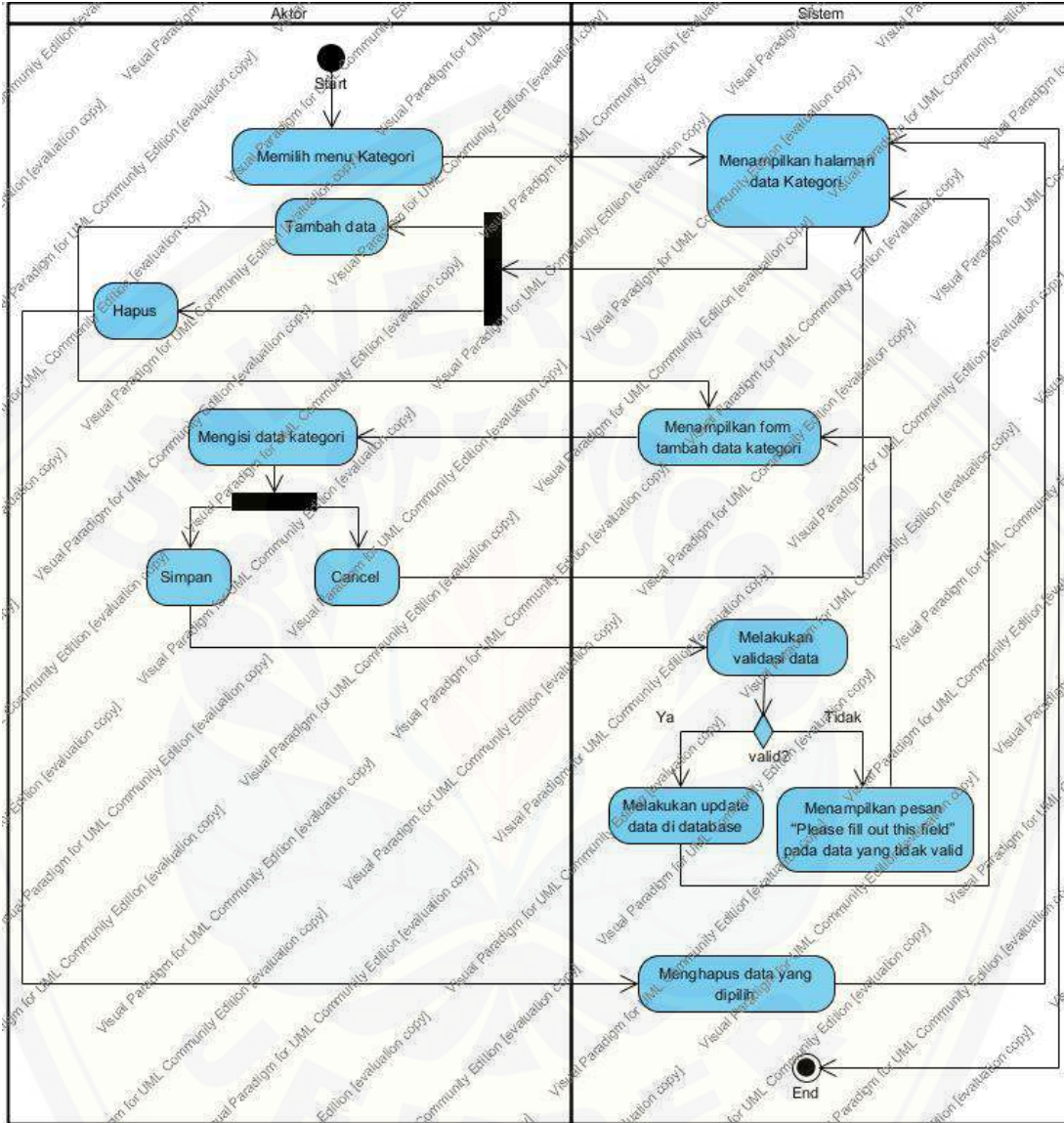
Gambar B.3. Activity diagram manajemen data pembelian

Lampiran B.4. Activity diagram manajemen data barang



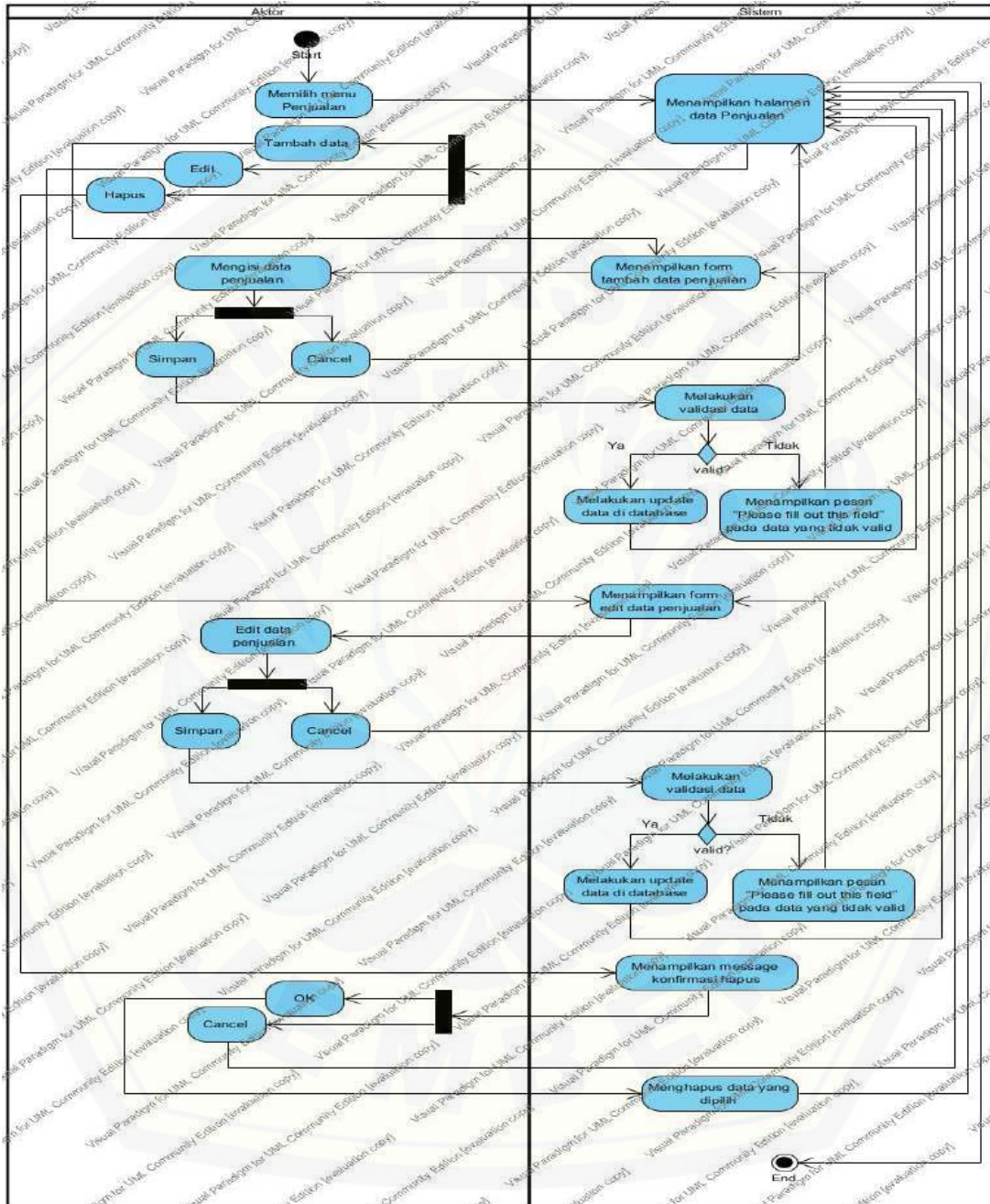
Gambar B.4. Activity diagram manajemen data barang

Lampiran B.6. Activity diagram manajemen data kategori



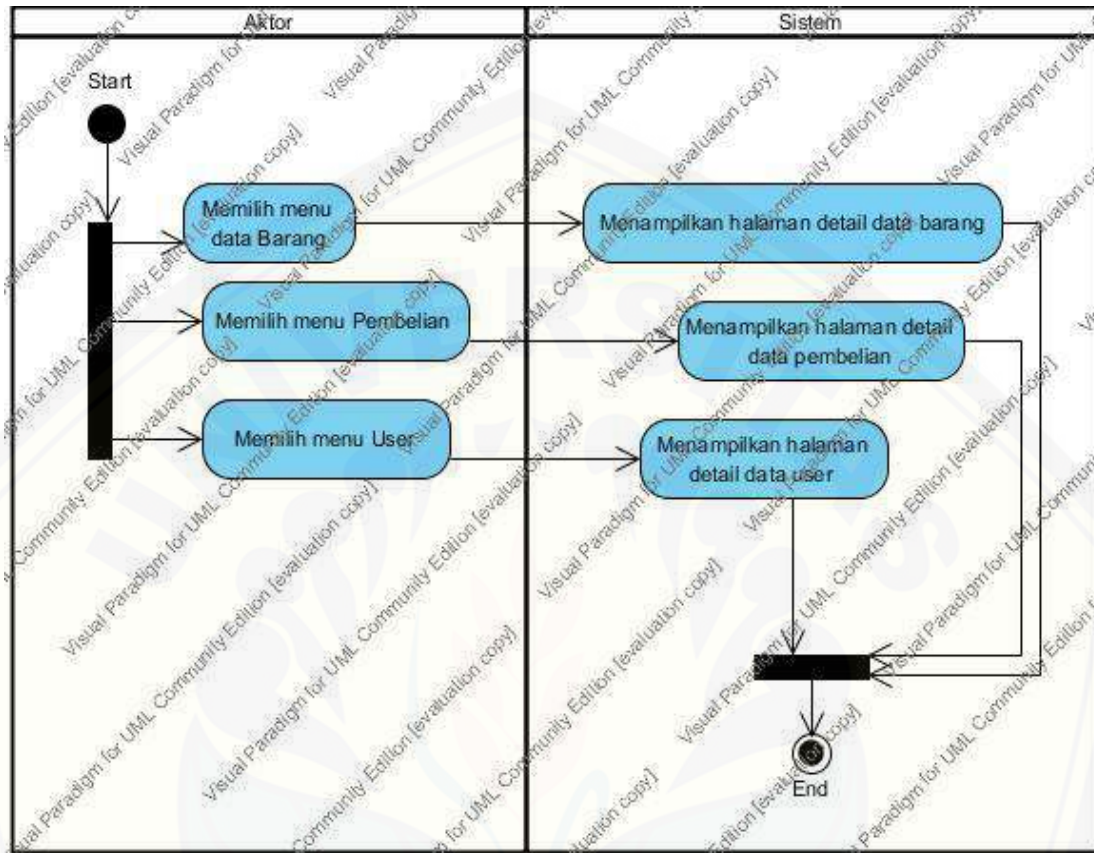
Gambar B.6. Activity diagram manajemen data kategori

Lampiran B.7. Activity diagram manajemen data penjualan



Gambar B.7. Activity diagram manajemen data penjualan

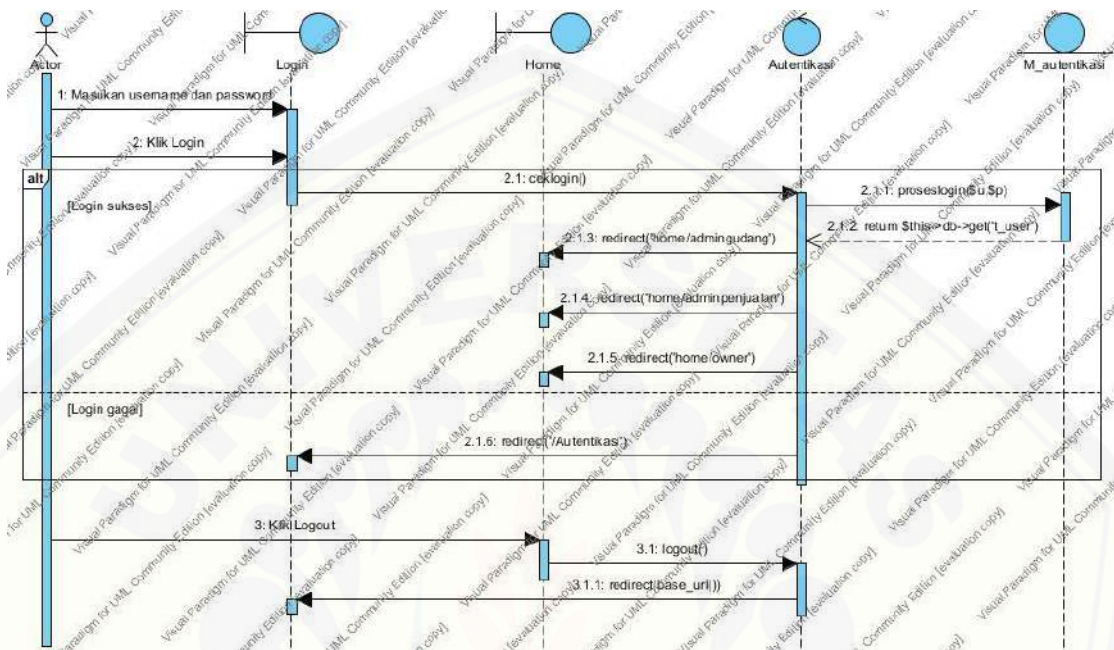
Lampiran B.8. Activity diagram melihat data transaksi barang



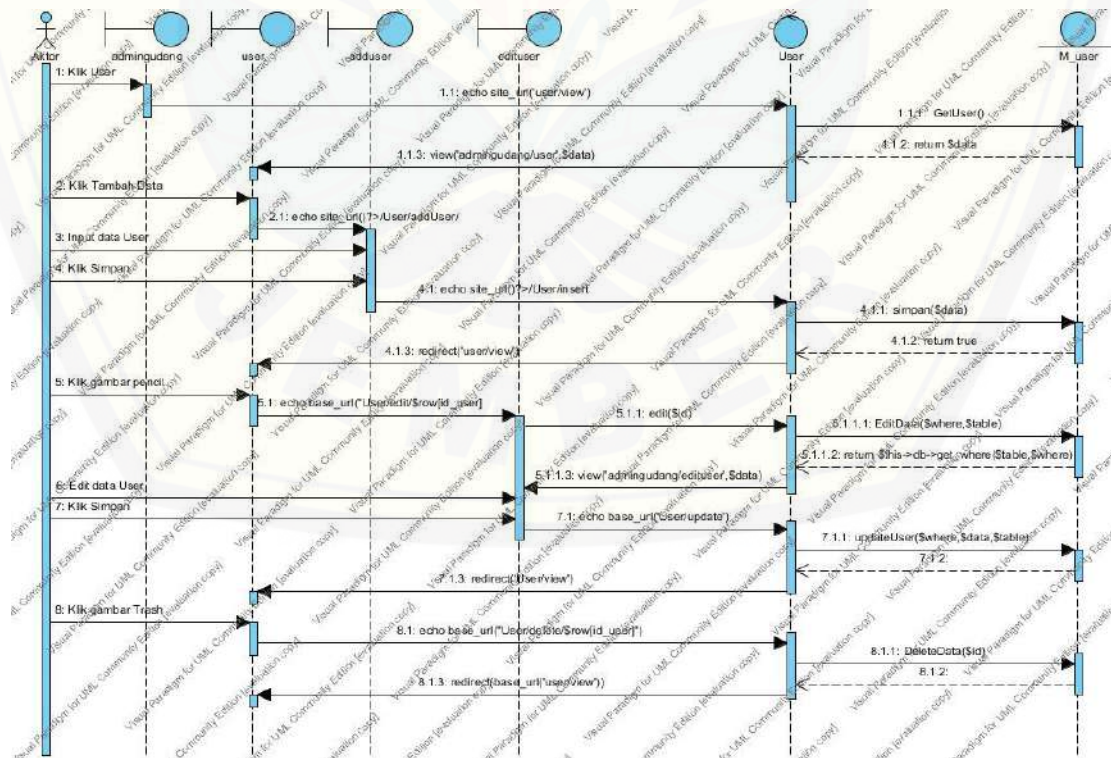
Gambar B.8. Activity diagram melihat data transaksi barang

LAMPIRAN C. SEQUENCE DIAGRAM

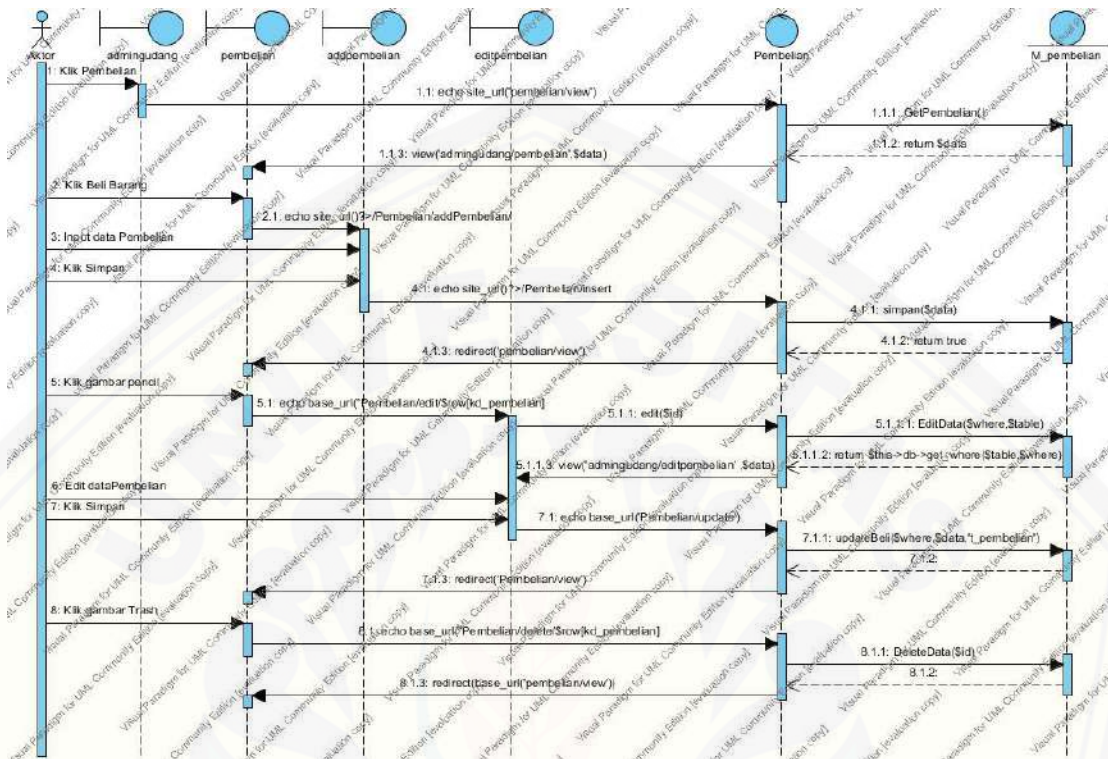
Lampiran C.1. Sequence Diagram Login



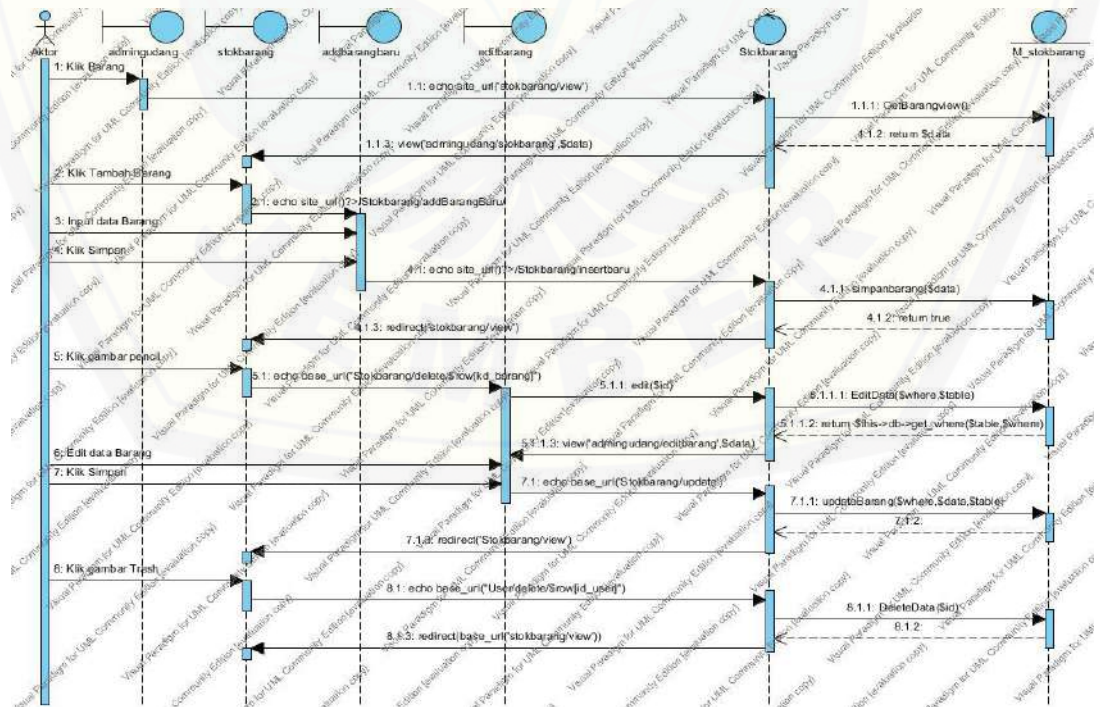
Lampiran C.2. Sequence Diagram Manajemen Data User



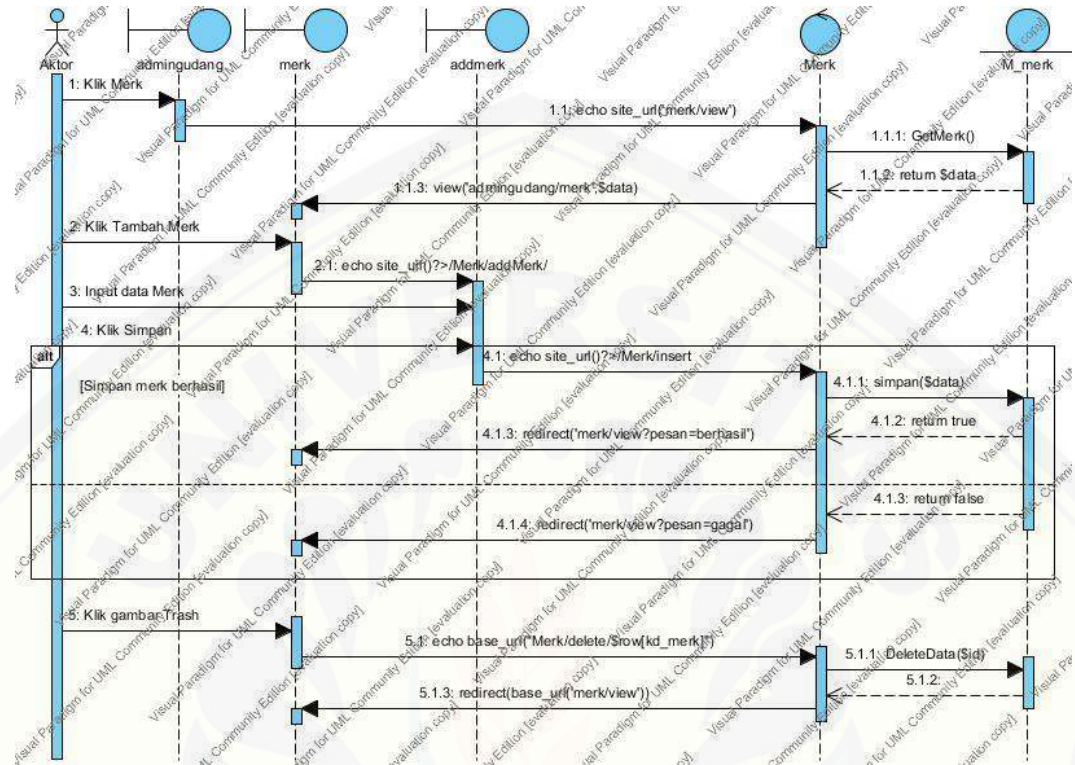
Lampiran C.3. Sequence Diagram Manajemen Data Pembelian



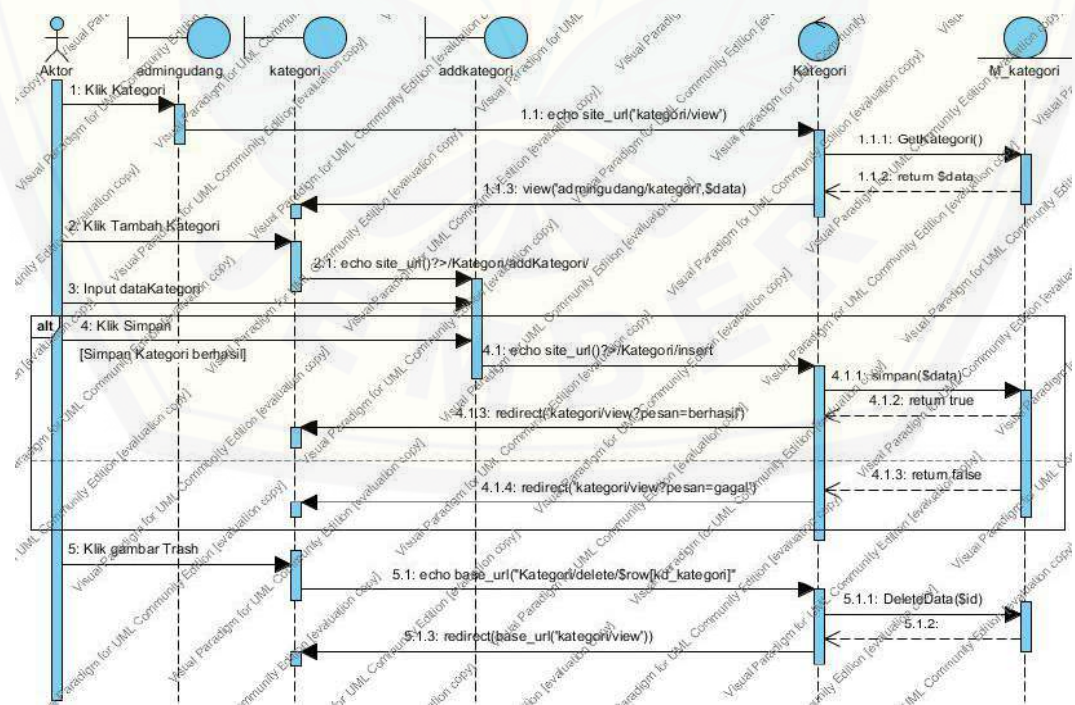
Lampiran C.4. Sequence Diagram Manajemen Data Barang



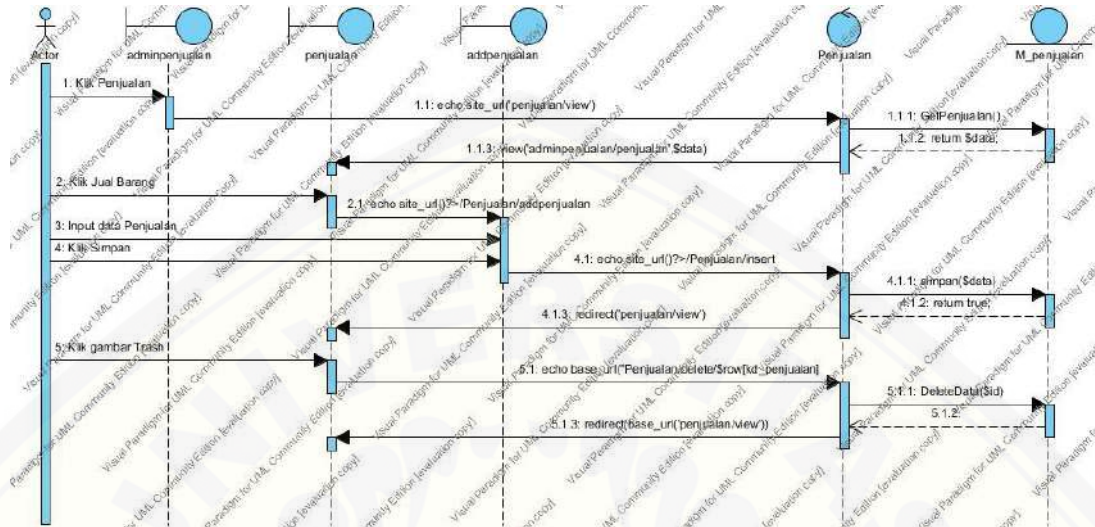
Lampiran C.5. Sequence Diagram Manajemen Data Merk



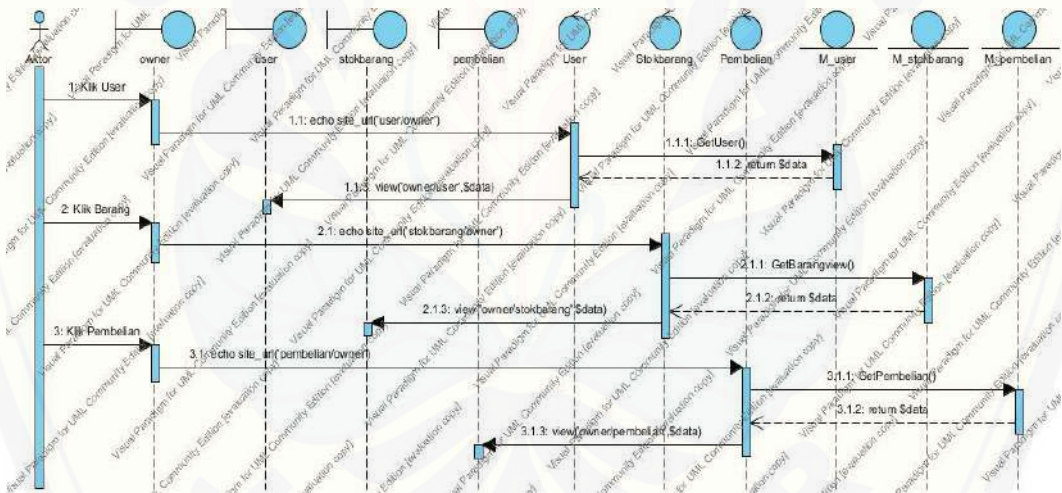
Lampiran C.6. Sequence Diagram Manajemen Data Kategori



Lampiran C.7. Sequence Diagram Manajemen Data Penjualan



Lampiran C.8. Sequence Diagram Melihat Data Transaksi Barang



LAMPIRAN D. PENULISAN KODE PROGRAM

Lampiran D.1. Autentikasi (*login*)

```
4     function __construct(){
5         parent::__construct();
6         $this->load->model('m_authentikasi');
7         $this->load->library('form_validation');
8     }
9     public function index(){
10        $this->load->view('login');
11    }
12    public function ceklogin(){
13        if(isset($_POST['login'])){
14            $u=$_this->input->post('username',true);
15            $p=$_this->input->post('password',true);
16            $cek=$_this->m_authentikasi->proseslogin($u,$p);
17            $ini = $cek->num_rows();
18            if($ini > 0){
19                $level;
20                foreach($cek->result_array() as $d){
21                    $level = $d['jabatan'];
22                }
23                if($level == 'admingudang'){
24                    redirect('home/admingudang');
25                } else if ($level == 'adminpenjualan') {
26                    redirect('home/adminpenjualan');
27                } else if ($level == 'owner') {
28                    redirect('home/owner');
29                }
30            }else{
31                redirect('/Autentikasi');
32            }
33        }
```

Lampiran D.2. Autentikasi (*logout*)

```
1     public function logout(){
2         $this->session->sess_destroy();
3         redirect(base_url());
4     }
```

Lampiran D.3. Manajemen Data User

```
1 public function view(){
2     $data['user'] = $this->m_user->GetUser();
3     $this->load->view('admingudang/header');
4     $this->load->view('admingudang/user',$data);
5     $this->load->view('footer');
6 }
7 public function owner(){
8     $data['user'] = $this->m_user->GetUser();
9     $this->load->view('owner/header');
10    $this->load->view('owner/user',$data);
11    $this->load->view('footer');
12 }
13 public function addUser(){
14    $this->load->view('admingudang/header');
15    $this->load->view('admingudang/adduser');
16    $this->load->view('footer');
17 }
18 }
19 public function insert(){
20    $data['nama'] = $this->input->post('nama');
21    $data['alamat'] = $this->input->post('alamat');
22    $data['jabatan'] = $this->input->post('jabatan');
23    $data['telepon'] = $this->input->post('telepon');
24    $data['email'] = $this->input->post('email');
25    $data['username'] = $this->input->post('username');
26    $data['password'] = $this->input->post('password');
27    $input = $this->m_user->simpan($data);
28    if($input == true){
29        redirect('user/view');
30    }
31 }
32 public function edit($id){
33    $where = array('id_user' => $id);
34    $data['user'] = $this->m_user->EditData($where,'t_user')->result();
35    $this->load->view('admingudang/header');
36    $this->load->view('admingudang/edituser',$data);
37    $this->load->view('footer');
38 }
39 public function update(){
40    $id = $this -> input->post('id_user');
41    $nama= $this -> input->post('nama');
42    $alamat= $this -> input->post('alamat');
43    $jabatan= $this -> input->post('jabatan');
44    $telepon= $this -> input->post('telepon');
45    $email= $this -> input->post('email');
46    $username= $this -> input->post('username');
47    $password= $this -> input->post('password');
48    $data = array(
49        'nama' => $nama,
50        'alamat' => $alamat,
51        'jabatan'=> $jabatan,
52        'telepon' => $telepon,
53        'email' => $email,
54        'username' => $username,
55        'password' => $password
56    );
```



```

57     $where = array(
58         'id_user' => $id
59     );
60     $this->m_user->updateUser($where,$data,'t_user');
61     redirect('User/view');
62 }
63 public function delete($id){
64     $this->m_user->DeleteData($id);
65     redirect(base_url('user/view'));
66 }
67 }

```

Lampiran D.4. Manajemen Data Pembelian

```

1  <?php
2  Class Pembelian extends CI_Controller {
3      function __construct()
4      {
5          parent::__construct();
6          $this->load->model('m_pembelian');
7      }
8      public function view(){
9          $data['pembelian'] = $this->m_pembelian->GetPembelian();
10         $this->load->view('admingudang/header');
11         $this->load->view('admingudang/pembelian',$data);
12         $this->load->view('footer');
13     }
14     public function addPembelian(){
15         $data['pembelian'] = $this->m_pembelian->GetPembelian();
16         $this->load->view('admingudang/header');
17         $this->load->view('admingudang/addpembelian',$data);
18         $this->load->view('footer');
19     }
20     public function owner(){
21         $data['pembelian'] = $this->m_pembelian->GetPembelian();
22         $this->load->view('owner/header');
23         $this->load->view('owner/pembelian',$data);
24         $this->load->view('footer');
25     }
26     public function insert(){
27         $data['tgl_pembelian'] = $this->input->post('tgl_pembelian');
28         $data['kd_barang'] = $this->input->post('kd_barang');
29         $data['jumlah_beli'] = $this->input->post('jumlah_beli');
30         $data['harga_beli'] = $this->input->post('harga_beli');
31         $data['harga_jual'] = $this->input->post('harga_jual');
32         $input = $this->m_pembelian->simpan($data);
33         if($input == true){
34             redirect('pembelian/view');
35         }
36     }

```



```

37     public function edit($id){
38         $where = array('kd_pembelian' => $id);
39         $data['pembelian'] = $this->m_pembelian->EditData($where, 't_pembelian')->result();
40         $this->load->view('admingudang/header');
41         $this->load->view('admingudang/editpembelian', $data);
42         $this->load->view('footer');
43     }
44     public function update(){
45         $id = $this -> input->post('kd_pembelian');
46         $tgl_pembelian= $this -> input->post('tgl_pembelian');
47         $kd_barang= $this -> input->post('kd_barang');
48         $jumlah_beli= $this -> input->post('jumlah_beli');
49         $harga_beli= $this -> input->post('harga_beli');
50         $harga_jual= $this -> input->post('harga_jual');
51         $data = array(
52             'tgl_pembelian' => $tgl_pembelian,
53             'kd_barang' => $kd_barang,
54             'jumlah_beli' => $jumlah_beli,
55             'harga_beli' => $harga_beli,
56             'harga_jual'=> $harga_jual
57         );
58         $where = array(
59             'kd_pembelian' => $id
60         );
61         $this->m_pembelian->updateBeli($where, $data, 't_pembelian');
62         redirect('Pembelian/view');
63     }
64     public function delete($id){
65         $this->m_pembelian->DeleteData($id);
66         redirect(base_url('pembelian/view'));
67     }
68 }
69 ?>

```

Lampiran D.5. Manajemen Data Barang

```

1  <?php
2  Class Stokbarang extends CI_Controller {
3      function __construct()
4      {
5          parent::__construct();
6          $this->load->model('m_stokbarang');
7      }
8      public function view(){
9          $data['stokbarang'] = $this->m_stokbarang->GetBarangview();
10         //$data['isi_kategori'] = $this->get_category();
11         $this->load->view('admingudang/header');
12         $this->load->view('admingudang/stokbarang', $data);
13         $this->load->view('footer');
14     }
15     public function pj(){
16         $this->load->view('adminpenjualan/header');
17         $this->load->view('adminpenjualan/stokbarang');
18         $this->load->view('footer');
19     }
20     public function owner(){
21         $data['stokbarang'] = $this->m_stokbarang->GetBarangview();
22         $this->load->view('owner/header');
23         $this->load->view('owner/stokbarang', $data);
24         $this->load->view('footer');
25     }

```

```
26     public function insertbaru(){
27         $data['kd_kategori'] = $this->input->post('kategori');
28         $data['kd_merk'] = $this->input->post('merk');
29         $data['type'] = $this->input->post('type');
30         $input = $this->m_stokbarang->simpanbarang($data);
31         if($input == true){
32             redirect('stokbarang/view');
33         }
34     }
35     public function addBarangBaru(){
36         $data['kategori'] = $this->m_stokbarang->GetKategori();
37         $data['merk'] = $this->m_stokbarang->GetMerk();
38         $this->load->view('admingudang/header');
39         $this->load->view('admingudang/addbarangbaru',$data);
40         $this->load->view('footer');
41     }
42     public function edit($id){
43         $where = array('kd_barang' => $id);
44         $data['stokbarang'] = $this->m_stokbarang->EditData($where,'t_barang')->result();
45         $this->load->view('admingudang/header');
46         $this->load->view('admingudang/editbarang',$data);
47         $this->load->view('footer');
48     }
49     public function update(){
50         $id = $this -> input->post('kd_barang');
51         $kategori= $this -> input->post('kategori');
52         $merk= $this -> input->post('merk');
53         $type= $this -> input->post('type');
54         $data = array(
55             'kategori' => $kategori,
56             'merk' => $merk,
57             'type'=> $type
58         );
59         $where = array(
60             'kd_barang' => $id
61         );
62         $this->m_stokbarang->updateBarang($where,$data,'t_barang');
63         redirect('Stokbarang/view');
64     }
65     public function delete($id){
66         $this->m_stokbarang->DeleteData($id);
67         redirect(base_url('stokbarang/view'));
68     }
69 }
70 ?>
```

Lampiran D.6. Manajemen Data Merk

```
1 <?php
2 Class Merk extends CI_Controller {
3     function __construct()
4     {
5         parent::__construct();
6         $this->load->model('m_merk');
7     }
8     public function view(){
9         $data['merk'] = $this->m_merk->GetMerk();
10        $this->load->view('admingudang/header');
11        $this->load->view('admingudang/merk',$data);
12        $this->load->view('footer');
13    }
14
15    public function addMerk(){
16        $this->load->view('admingudang/header');
17        $this->load->view('admingudang/addmerk');
18        $this->load->view('footer');
19    }
20
21    public function insert(){
22        $data['kd_merk'] = $this->input->post('kd_merk');
23        $data['merk'] = $this->input->post('merk');
24        $input = $this->m_merk->simpan($data);
25        if($input == true){
26            redirect('merk/view?pesan=berhasil');
27        }else{
28            redirect('merk/view?pesan=gagal');
29        }
30    }
31
32    public function delete($id){
33        $this->m_merk->DeleteData($id);
34        redirect(base_url('merk/view'));
35    }
36 }
37 ?>
```


Lampiran D.7. Manajemen Data Kategori

```
1 <?php
2 Class Kategori extends CI_Controller {
3     function __construct()
4     {
5         parent::__construct();
6         $this->load->model('m_kategori');
7     }
8     public function view(){
9         $data['kategori'] = $this->m_kategori->GetKategori();
10        $this->load->view('admingudang/header');
11        $this->load->view('admingudang/kategori',$data);
12        $this->load->view('footer');
13    }
14
15    public function addKategori(){
16        $this->load->view('admingudang/header');
17        $this->load->view('admingudang/addkategori');
18        $this->load->view('footer');
19    }
20
21    public function insert(){
22        $data['kd_kategori'] = $this->input->post('kd_kategori');
23        $data['kategori'] = $this->input->post('kategori');
24        $input = $this->m_kategori->simpan($data);
25        if($input == true){
26            redirect('kategori/view?pesan=berhasil');
27        }else{
28            redirect('kategori/view?pesan=gagal');
29        }
30    }
31
32    public function delete($id){
33        $this->m_kategori->DeleteData($id);
34        redirect(base_url('kategori/view'));
35    }
36 }
37 ?>
```

Lampiran D.8. Manajemen Data Penjualan

```

1  <?php
2  Class Penjualan extends CI_Controller {
3      function __construct(){
4          parent::__construct();
5          $this->load->model('m_penjualan');
6      }
7      public function view(){
8          $data['penjualan'] = $this->m_penjualan->GetPenjualan();
9          $this->load->view('adminpenjualan/header');
10         $this->load->view('adminpenjualan/penjualan',$data);
11         $this->load->view('footer');
12     }
13     public function pj(){
14         $this->load->view('adminpenjualan/header');
15         $this->load->view('adminpenjualan/penjualan');
16         $this->load->view('footer');
17     }
18     public function addPenjualan(){
19         $this->load->view('adminpenjualan/header');
20         $this->load->view('adminpenjualan/addpenjualan');
21         $this->load->view('footer');
22     }
23     public function insert(){
24         $data['kd_penjualan'] = $this->input->post('kd_penjualan');
25         $data['kd_barang'] = $this->input->post('kd_barang');
26         $data['harga_jual'] = $this->input->post('harga_jual');
27         $data['jumlah_jual'] = $this->input->post('jumlah_jual');
28         $input = $this->m_penjualan->simpan($data);
29         if($input == true){
30             redirect('penjualan/view');
31         }
32     }
33     public function delete($id){
34         $this->m_penjualan->DeleteData($id);
35         redirect(base_url('penjualan/view'));
36     }
37 }
38 ?>

```

Lampiran D.9. view User oleh Owner

```

7  public function owner(){
8      $data['user'] = $this->m_user->GetUser();
9      $this->load->view('owner/header');
10     $this->load->view('owner/user',$data);
11     $this->load->view('footer');
12 }

```

Lampiran D.10. Data Transaksi view Barang

```

19 }
20 public function owner(){
21     $data['stokbarang'] = $this->m_stokbarang->GetBarangview();
22     $this->load->view('owner/header');
23     $this->load->view('owner/stokbarang',$data);
24     $this->load->view('footer');
25 }

```


Lampiran D.11. Data Transaksi *view* Pembelian

```

20 public function owner(){
21     $data['pembelian'] = $this->m_pembelian->GetPembelian();
22     $this->load->view('owner/header');
23     $this->load->view('owner/pembelian',$data);
24     $this->load->view('footer');
25 }

```

LAMPIRAN E. PENGUJIAN *WHITE BOX*

Lampiran E.1. Autentikasi (*login*)

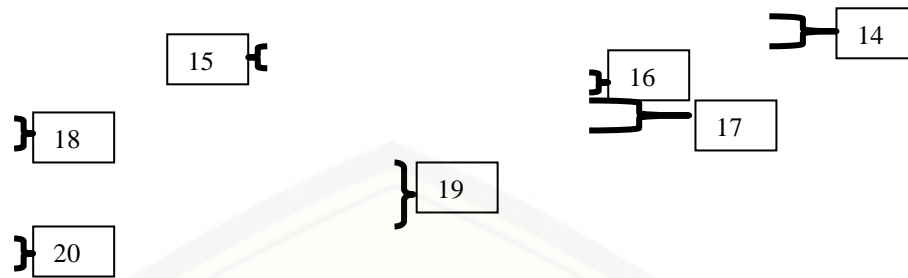
Pengujian white box pada fitur autentikasi (*login*) adalah sebagai berikut:

1. Listing Program

```

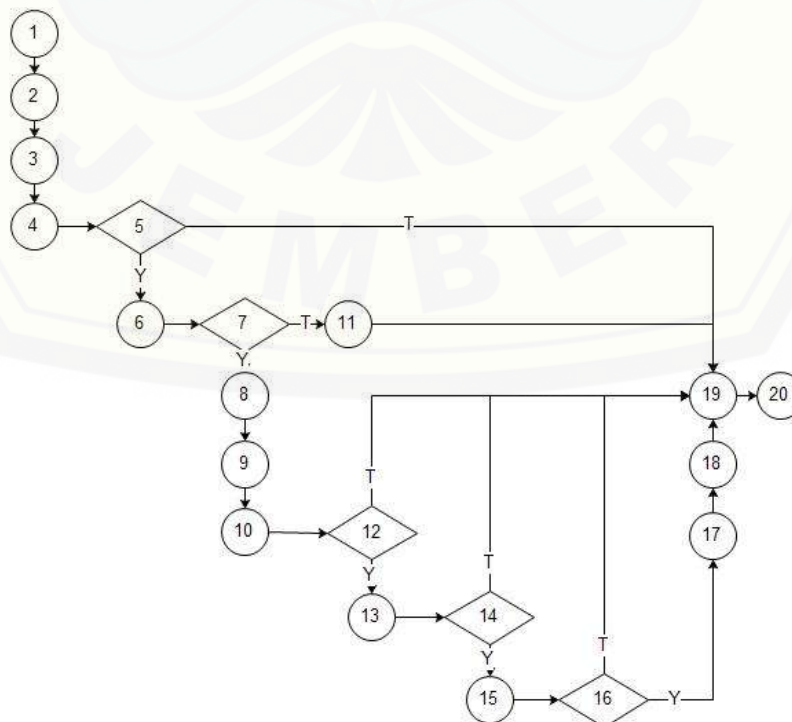
9     public function index(){ } 1
10     $this->load->view('login'); } 2
11 } } 3
12     public function ceklogin(){ } 4
13     if(isset($_POST['login'])){ } 5
14         $u=$_this->input->post('username',true);
15         $p=$_this->input->post('password',true);
16         $cek=$this->m_authentikasi->proseslogin($u,$p);
17         $sini = $cek->num_rows();
18         if($sini > 0){ } 7
19             $level;
20             foreach($cek->result_array() as $d){ } 9
21                 $level = $d['jabatan']; } 10
22             } } 11
23             if($level == 'admingudang'){
24                 redirect('home/admingudang'); } 13
25             } else if ($level == 'adminpenjualan') {
26                 redirect('home/adminpenjualan');
27             } else if ($level == 'owner') {
28                 redirect('home/owner');
29         }
30     }else{
31         redirect('/Autentikasi');

```



2. Grafik Alir

Grafik Alir fitur Autentikasi (*login*). Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.1.



Gambar E.1. Grafik Alir Autentikasi (*login*)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Autentikasi (*login*) dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 24 - 20 + 2 = 6$$

4. Basis Set

Basis set dari fitur Autentikasi (*login*) menghasilkan 6 jalur yaitu :

Jalur 1 : 1-2-3-4-5-19-20

Jalur 2 : 1-2-3-4-5-6-7-11-19-20

Jalur 3 : 1-2-3-4-5-6-7-8-9-10-12-19-20

Jalur 4 : 1-2-3-4-5-6-7-8-9-10-12-13-14-19-20

Jalur 5 : 1-2-3-4-5-6-7-8-9-10-12-1-14-15-16-19-20

Jalur 6 : 1-2-3-4d-5-6-7-8-9-10-12-13-14-15-16-17-18-19-20

5. Test Case

Tabel E.1. Autentikasi (*login*) Jalur 1

Test case	Sistem menampilkan form <i>login</i> , kemudian <i>user</i> tidak mengisi <i>username</i> dan <i>password</i> dan menekan tombol <i>login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>login</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	Autentikasi.php

Tabel E.2. Autentikasi (*login*) Jalur 2

Test case	<i>User</i> mengisi <i>username</i> dan <i>password</i> yang salah kemudian klik <i>login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>login</i>
Hasil pengujian	Benar
Jalur	Jalur 2

File	Autentikasi.php
------	-----------------

Tabel E.3. Autentikasi (*login*) Jalur 3

Test case	User mengisi <i>username</i> dan <i>password</i> dan sistem mengecek hak akses <i>user</i> admin gudang dan tidak tersedia
Target yang diharapkan	Sistem menampilkan halaman <i>login</i>
Hasil pengujian	Benar
Jalur	Jalur 3
File	Autentikasi.php

Tabel E.4. Autentikasi (*login*) Jalur 4

Test case	User mengisi <i>username</i> dan <i>password</i> yang sesuai dengan hak akses admin gudang
Target yang diharapkan	Sistem menampilkan hak akses admin gudang
Hasil pengujian	Benar
Jalur	Jalur 4
File	Autentikasi.php

Tabel E.5. Autentikasi (*login*) Jalur 5

Test case	User mengisi <i>username</i> dan <i>password</i> yang tidak sesuai hak akses
Target yang diharapkan	Sistem menampilkan halaman <i>login</i>
Hasil pengujian	Benar
Jalur	Jalur 5
File	Autentikasi.php

Tabel E.6. Autentikasi (*login*) Jalur 6

Test case	User mengisi <i>username</i> dan <i>password</i> yang sesuai hak akses
Target yang diharapkan	Sistem menampilkan halaman sesuai hak akses
Hasil pengujian	Benar
Jalur	Jalur 6
File	Autentikasi.php

Lampiran E.2. Logout

Pengujian *white box* pada fitur *logout* adalah sebagai berikut:

1. Listing Program

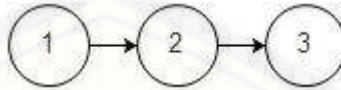
```

1 public function logout() { } 1
2     $this->session->sess_destroy();
3     redirect(base_url());
4 } 3

```


2. Grafik Alir

Grafik Alir fitur *logout*. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.2.



Gambar E.3. Grafik Alir *logout*

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *logout* dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur Autentikasi (*logout*) menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3

5. Test Case

Tabel E.7. Autentikasi (*logout*) Jalur 1

Test case	User menekan tombol <i>logout</i>
Target yang diharapkan	Sistem kembali kehalaman <i>login</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	Autentikasi.php

Lampiran E.3. Menu User

Pengujian *white box* pada fitur menu *user* adalah sebagai berikut:

1. Listing Program

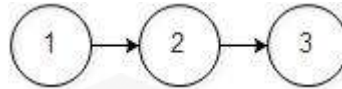
```

1 public function view() { } 1
2 { $data['user'] = $this->m_user->GetUser();
3   $this->load->view('admingudang/header');
4   $this->load->view('admingudang/user', $data);
5   $this->load->view('footer');
6 } } 3
  
```

2. Grafik Alir

Grafik Alir fitur menu *user*. Grafik alir tersebut di gambarkan dengan simpul

dan garis dapat di lihat pada Gambar E.3.



Gambar E.2. Grafik Alir Menu *User*

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur menu *user* dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. *Basis Set*

Basis set dari fitur menu *user* menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3

5. *Test Case*

Tabel E.8. Menu *User* Jalur 1

Test case	<i>User</i> menekan menu <i>user</i>
Target yang diharapkan	Sistem menampilkan halaman <i>user</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	User.php

Lampiran E.4. Tambah *User*

Pengujian *white box* pada fitur tambah *user* adalah sebagai berikut:

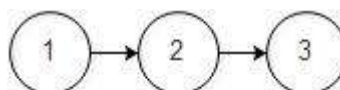
1. *Listing Program*

```

13 public function addUser() { } 1
14     $this->load->view('admingudang/header');
15     $this->load->view('admingudang/adduser'); } 2
16     $this->load->view('footer');
17 } 3
18
    
```

2. Grafik Alir

Grafik Alir fitur tambah *user*. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.4.



Gambar E.4. Grafik Alir Tambah *User*

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur tambah *user* dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. *Basis Set*

Basis set dari fitur tambah *user* menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3

5. *Test Case*

Tabel E.9. Tambah *User* Jalur 1

Test case	<i>User</i> menekan tombol tambah data pada halaman <i>user</i>
Target yang diharapkan	Sistem menampilkan halaman tambah <i>user</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	adduser.php

Lampiran E.5. *Insert User*

Pengujian *white box* pada fitur *insert user* adalah sebagai berikut:

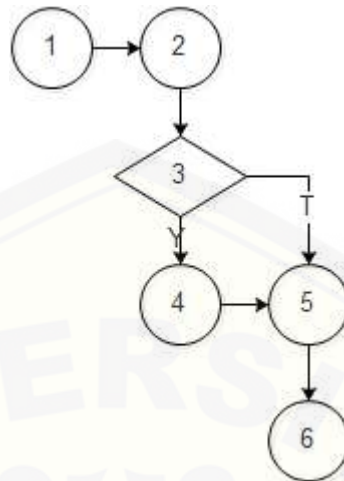
1. *Listing Program*

```

19     public function insert() { 1
20         $data['nama'] = $this->input->post('nama');
21         $data['alamat'] = $this->input->post('alamat');
22         $data['jabatan'] = $this->input->post('jabatan');
23         $data['telepon'] = $this->input->post('telepon');
24         $data['email'] = $this->input->post('email');
25         $data['username'] = $this->input->post('username');
26         $data['password'] = $this->input->post('password');
27         $input = $this->input->simpan($data);
28         if($input == true) { 3
29             redirect('user/view'); 4
30         } 5
31     } 6
    
```

2. *Grafik Alir*

Grafik Alir fitur *insert user*. *Grafik alir* tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.5.



Gambar E.5. Grafik Alir *Insert User*

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *insert user* dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 6 - 6 + 2 = 2$$

4. *Basis Set*

Basis set dari fitur *insert user* menghasilkan 2 jalur yaitu :

Jalur 1 = 1-2-3-4-5-6

Jalur 2 = 1-2-3-5-6

5. *Test Case*

Tabel E.10. *Insert User Jalur 1*

Test case	<i>User</i> mengisi form data <i>insertuser</i> dan menekan tombol simpan
Target yang diharapkan	<i>input data</i> berhasil
Hasil pengujian	Benar
Jalur	Jalur 1
File	<code>adduser.php</code>

Tabel E.11. *Insert User Jalur 2*

Test case	<i>User</i> mengisi form <i>insert user</i> tidak lengkap dan klik simpan
Target yang diharapkan	Sistem menampilkan “ <i>please fill out this field</i> ” pada kolom yang belum terisi
Hasil pengujian	Benar

Jalur	Jalur 2
File	adduser.php

Lampiran E.6. Edit User

Pengujian *white box* pada fitur *edit user* adalah sebagai berikut:

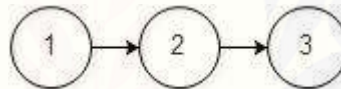
1. Listing Program

```

32 public function edit($id) { } 1
33     $where = array('id_user' => $id);
34     $data['user'] = $this->m_user->EditData($where, 't_user')->result();
35     $this->load->view('admingudang/header');
36     $this->load->view('admingudang/edituser', $data);
37     $this->load->view('footer');
38 }
    
```

2. Grafik Alir

Grafik Alir fitur *edit user*. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.6.



Gambar E.6. Grafik Alir *Edit User*

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *edit user* dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur *edit user* menghasilkan 1 jalur yaitu :

$$\text{Jalur 1} = 1-2-3$$

5. Test Case

Tabel E.12. *Edit User* Jalur 1

Test case	User menekan tombol pensil pada salah satu kolom data <i>edit user</i>
Target yang diharapkan	Menampilkan form <i>edit user</i> sesuai kolom yang dipilih
Hasil pengujian	Benar
Jalur	Jalur 1

File	User.php
------	----------

Lampiran E.7. Update User

Pengujian *white box* pada fitur *update user* adalah sebagai berikut:

1. Listing Program

```

39     public function update() { 1
40         $id = $this->input->post('id_user');
41         $nama= $this->input->post('nama');
42         $alamat= $this->input->post('alamat');
43         $jabatan= $this->input->post('jabatan');
44         $telepon= $this->input->post('telepon');
45         $email= $this->input->post('email');
46         $username= $this->input->post('username');
47         $password= $this->input->post('password');
48         $data = array(
49             'nama' => $nama,
50             'alamat' => $alamat,
51             'jabatan'=> $jabatan,
52             'telepon' => $telepon,
53             'email' => $email,
54             'username' => $username,
55             'password' => $password
56         ); 4
57         $where = array(
58             'id_user' => $id 5
59         ); 6
60         $this->_user->updateUser($where, $data, 't_user');
61         redirect('User/view');
62     } 8

```

2. Grafik Alir

Grafik Alir fitur *update user*. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.7.



Gambar E.7. Grafik Alir *update User*

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *update user* dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur *update user* menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3-4-5-6-7-8

5. *Test Case*Tabel E.13. *Update User Jalur 1*

Test case	<i>User</i> mengubah data <i>user</i> dan klik tombol simpan
Target yang diharapkan	Sistem mengupdate data <i>user</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	User.php

Lampiran E.8. Hapus User

Pengujian white box pada fitur hapus *user* adalah sebagai berikut:

1. *Listing Program*

```

63 public function delete($id) { 1
64     $this->m_user->DeleteData($id);
65     redirect(base_url('user/view')); 2
66 } 3

```

2. *Grafik Alir*

Grafik Alir fitur hapus *user*. Grafik alir tersebut di gambarkan dengan simpel dan garis dapat di lihat pada Gambar E.8.

Gambar E.8. *Grafik Alir Hapus User*3. *Kompleksitas siklomatis*

Kompleksitas siklomatis dari fitur hapus *user* dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. *Basis Set*

Basis set dari fitur hapus *user* menghasilkan 1 jalur yaitu :

$$\text{Jalur 1} = 1-2-3$$

5. *Test Case*Tabel E.14. *Hapus User Jalur 1*

Test case	User menekan tombol hapus
Target yang diharapkan	Data terhapus
Hasil pengujian	Benar
Jalur	Jalur 1
File	User.php

Lampiran E.9. Menu Pembelian

Pengujian *white box* pada fitur menu pembelian adalah sebagai berikut:

1. Listing Program

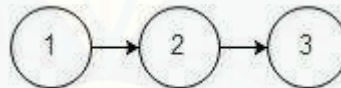
```

8 public function view() {
9     $data['pembelian'] = $this->m_pembelian->GetPembelian();
10    $this->load->view('admingudang/header');
11    $this->load->view('admingudang/pembelian', $data);
12    $this->load->view('footer');
13 }

```

2. Grafik Alir

Grafik Alir fitur menu pembelian. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.9.



Gambar E.9. Grafik Alir Menu Pembelian

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur menu pembelian dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur menu pembelian menghasilkan 1 jalur yaitu :

$$\text{Jalur 1} = 1-2-3$$

5. Test Case

Tabel E.15. Menu Pembelian Jalur 1

Test case	User menekan menu pembelian
Target yang diharapkan	Sistem menampilkan halaman pembelian
Hasil pengujian	Benar
Jalur	Jalur 1
File	Pembelian.php

Lampiran E.10. Tambah Pembelian

Pengujian *white box* pada fitur tambah pembelian adalah sebagai berikut:

1. Listing Program

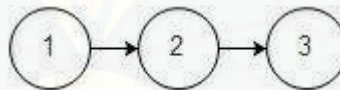
```

14 public function addPembelian() { 1
15     $data['pembelian'] = $this->m_pembelian->GetPembelian();
16     $this->load->view('admingudang/header');
17     $this->load->view('admingudang/addpembelian', $data);
18     $this->load->view('footer');
19 } 3

```

2. Grafik Alir

Grafik Alir fitur tambah pembelian. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.10.



Gambar E.10. Grafik Alir Tambah Pembelian

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur tambah pembelian dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur tambah pembelian menghasilkan 1 jalur yaitu :

$$\text{Jalur 1} = 1-2-3$$

5. Test Case

Tabel E.16. Tambah Pembelian Jalur 1

Test case	User menekan tombol beli barang pada halaman pembelian
Target yang diharapkan	Sistem menampilkan halaman pembelian
Hasil pengujian	Benar
Jalur	Jalur 1
File	addpembelian.php

Lampiran E.11. *Insert* Pembelian

Pengujian *white box* pada fitur *insert* pembelian adalah sebagai berikut:

1. Listing Program

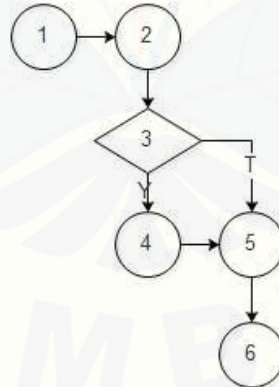
```

26 public function insert() { 1
27     $data['tgl_pembelian'] = $this->input->post('tgl_pembelian');
28     $data['kd_barang'] = $this->input->post('kd_barang');
29     $data['jumlah_beli'] = $this->input->post('jumlah_beli');
30     $data['harga_beli'] = $this->input->post('harga_beli');
31     $data['harga_jual'] = $this->input->post('harga_jual');
32     $input = $this->m_pembelian->simpan($data);
33     if($input == true) { 3
34         redirect('pembelian/view'); 4
35     } 5
36 } 6

```

2. Grafik Alir

Grafik Alir fitur *insert* pembelian. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.11.



Gambar E.11. Grafik Alir *Insert* Pembelian

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *insert* pembelian dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 6 - 6 + 2 = 2$$

4. *Basis Set*

Basis set dari fitur *insert* pembelian menghasilkan 2 jalur yaitu :

Jalur 1= 1-2-3-4-5-6

Jalur 2= 1-2-3-5-6

5. *Test Case*Tabel E.17. *Insert* Pembelian Jalur 1

Test case	User mengisi form data pembelian dan menekan tombol simpan
Target yang diharapkan	Input data berhasil
Hasil pengujian	Benar
Jalur	Jalur 1
File	addpembelian.php

Tabel E.18. *Insert* Pembelian Jalur 2

Test case	User mengisi form pembelian tidak lengkap dan klik simpan
Target yang diharapkan	Sistem menampilkan “ <i>please fill out this field</i> ” pada kolom yang belum trisi
Hasil pengujian	Benar
Jalur	Jalur 2
File	addpembelian.php

Lampiran E.12. *Edit* Pembelian

Pengujian *white box* pada fitur *edit* pembelian adalah sebagai berikut:

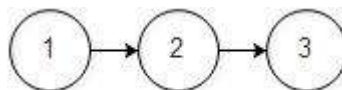
1. *Listing* Program

```

37 public function edit($id) 1
38     $where = array('kd_pembelian' => $id);
39     $data['pembelian'] = $this->m_pembelian->EditData($where, 't_pembelian')->result();
40     $this->load->view('admingudang/header');
41     $this->load->view('admingudang/editpembelian' , $data);
42     $this->load->view('footer');
43 
```

2. Grafik Alir

Grafik Alir fitur *edit* pembelian. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.12.



Gambar E.12. Grafik Alir *Edit* Pembelian

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *edit* pembelian dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. *Basis Set*

Basis set dari fitur *edit* pembelian menghasilkan 1 jalur yaitu :

$$\text{Jalur 1} = 1-2-3$$

5. *Test Case*

Tabel E.19. *Edit* Pembelian Jalur 1

Test case	User menekan tombol pensil pada salah satu kolom data pembelian
Target yang diharapkan	Menampilkan form <i>edit</i> pembelian
Hasil pengujian	Benar
Jalur	Jalur 1
File	Pembelian.php

Lampiran E.13. *Update* Pembelian

Pengujian white box pada fitur *update* pembelian adalah sebagai berikut:

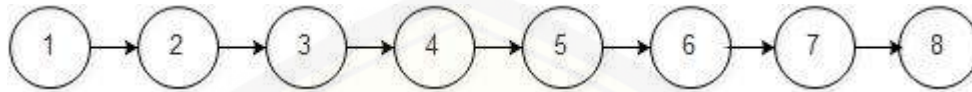
1. *Listing Program*

```

44 public function update() { 1
45     $id = $this->input->post('kd_pembelian');
46     $tgl_pembelian= $this->input->post('tgl_pembelian');
47     $kd_barang= $this->input->post('kd_barang');
48     $jumlah_beli= $this->input->post('jumlah_beli');
49     $harga_beli= $this->input->post('harga_beli');
50     $harga_jual= $this->input->post('harga_jual');
51     $data = array(
52         'tgl_pembelian' => $tgl_pembelian,
53         'kd_barang' => $kd_barang,
54         'jumlah_beli' => $jumlah_beli,
55         'harga_beli' => $harga_beli,
56         'harga_jual' => $harga_jual
57     ); 4
58     $where = array(
59         'kd_pembelian' => $id 5
60     ); 6
61     $this-> Pembelian->updateBeli($where,$data,'t_pembelian');
62     redirect('Pembelian/view');
63 } 8
    
```

2. Grafik Alir

Grafik Alir *update* pembelian. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.13.



Gambar E.13. Grafik Alir *Update* Pembelian

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *update* pembelian dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 7 - 8 + 2 = 1$$

4. *Basis Set*

Basis set dari fitur *update* pembelian menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3-4-5-6-7-8

5. *Test Case*

Tabel E.20. *Update* Pembelian Jalur 1

Test case	User mengubah data pembelian dan klik tombol simpan
Target yang diharapkan	Sistem mengupdate data pembelian
Hasil pengujian	Benar
Jalur	Jalur 1
File	Pembelian.php

Lampiran E.14. Hapus Pembelian

Pengujian *white box* pada hapus pembelian adalah sebagai berikut:

1. *Listing Program*

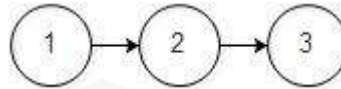
```

64     public function delete($id) { 1
65         2 { $this->m_pembelian->DeleteData($id);
66           redirect(base_url('pembelian/view'));
67         } 3
    
```

2. Grafik Alir

Grafik Alir hapus pembelian. Grafik alir tersebut di gambarkan dengan

simpul dan garis dapat di lihat pada Gambar E.14.



Gambar E.14. Grafik Alir Hapus Pembelian

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data Merk dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur hapus pembelian menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3

5. Test Case

Tabel E.21. Hapus Pembelian Jalur 1

Test case	User menghapus data pembelian
Target yang diharapkan	Data terhapus
Hasil pengujian	Benar
Jalur	Jalur 1
File	Pembelian.php

Lampiran E.15. Menu Barang

Pengujian white box pada fitur menu barang adalah sebagai berikut:

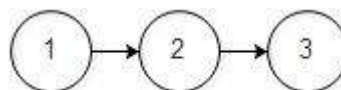
1. Listing Program

```

8 public function view() { 1
9     $data['stokbarang'] = $this->m_stokbarang->GetBarangview();
10    // $data['isi_kategori'] = $this->get_category();
11    $this->load->view('admingudang/header');
12    $this->load->view('admingudang/stokbarang', $data);
13    $this->load->view('footer');
14 } 3
    
```

2. Grafik Alir

Grafik Alir fitur menu barang. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.15.



Gambar E.15. Grafik Alir Menu Barang

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur menu barang dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur menu barang menghasilkan 1 jalur yaitu :

$$\text{Jalur 1} = 1-2-3$$

5. Test Case

Tabel E.22. Menu Barang Jalur 1

Test case	User menekan menu barang
Target yang diharapkan	Sistem menampilkan halaman barang
Hasil pengujian	Benar
Jalur	Jalur 1
File	Stokbarang.php

Lampiran E.16. Tambah Barang

Pengujian *white box* pada tambah barang adalah sebagai berikut:

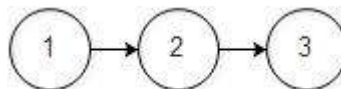
1. Listing Program

```

35     public function addBarangBaru() {
36         $data['kategori'] = $this->m_stokbarang->GetKategori();
37         $data['merk'] = $this->m_stokbarang->GetMerk();
38         $this->load->view('admingudang/header');
39         $this->load->view('admingudang/addbarangbaru', $data);
40         $this->load->view('footer');
41     }
    
```

2. Grafik Alir

Grafik Alir tambah barang. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.16.



Gambar E.16. Diagram Alir Tambah Barang

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari tambah barang dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. *Basis Set*

Basis set dari fitur tambah barang menghasilkan 1 jalur yaitu :

Jalur 1= 1-2-3

5. *Test Case*

Tabel E.23. Tambah Barang Jalur 1

Test case	User menekan tombol tambah data pada halaman barang
Target yang diharapkan	Sistem menampilkan halaman barang
Hasil pengujian	Benar
Jalur	Jalur 1
File	addBarangBaru.php

Lampiran E.17. Insert Barang

Pengujian *white box* pada fitur *insert* barang adalah sebagai berikut:

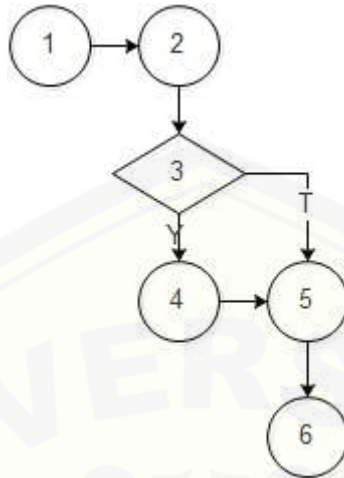
1. *Listing Program*

```

26 public function insertbaru() { 1
27     $data['kd_kategori'] = $this->input->post('kategori');
28     $data['kd_merk'] = $this->input->post('merk');
29     $data['type'] = $this->input->post('type');
30     $input = $this->m_stokbarang->simpanbarang($data);
31     if($input == true) {
32         redirect('stokbarang/view'); 3
33     }
34 } 4
    } 5
    
```

2. *Grafik Alir*

Grafik Alir fitur *insert* barang. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.17.

Gambar E.17. Grafik Alir *Insert* Barang

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *insert* barang dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 6 - 6 + 2 = 2$$

4. *Basis Set*

Basis set dari fitur *insert* barang menghasilkan 2 jalur yaitu :

Jalur 1 = 1-2-3-4-5-6

Jalur 2 = 1-2-3-5-6

5. *Test Case*Tabel E.24. *Insert* Barang Jalur 1

Test case	<i>User</i> mengisi form data barang dan menekan tombol simpan
Target yang diharapkan	<i>Input</i> data berhasil
Hasil pengujian	Benar
Jalur	Jalur 1
File	addBarangBaru.php

Tabel E.25. *Insert* Barang Jalur 2

Test case	<i>User</i> mengisi form barang tidak lengkap dan klik simpan
Target yang diharapkan	Sistem menampilkan " <i>please fill out this field</i> "

	pada kolom yang belum terisi
Hasil pengujian	Benar
Jalur	Jalur 2
File	addBarangBaru.php

Lampiran E.18. Edit Barang

Pengujian *white box* pada fitur *edit* barang adalah sebagai berikut:

1. Listing Program

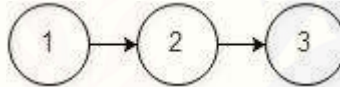
```

42 public function edit($id) { 1
43     $where = array('kd_barang' => $id);
44     $data['stokbarang'] = $this->m_stokbarang->EditData($where, 't_barang')->result();
45     $this->load->view('admingudang/header');
46     $this->load->view('admingudang/editbarang', $data);
47     $this->load->view('footer');
48 }

```

2. Grafik Alir

Grafik Alir fitur *edit* barang. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.18.



Gambar E.18. Grafik Alir *Edit* Barang

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *edit* barang dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur *edit* barang menghasilkan 1 jalur yaitu :

$$\text{Jalur 1} = 1-2-3$$

5. Test Case

Tabel E.26. *Edit* Barang Jalur 1

Test case	User menekan tombol pensil pada salah satu kolom data barang
Target yang diharapkan	Menampilkan form <i>edit</i> barang sesuai kolom yang dipilih
Hasil pengujian	Benar

Jalur	Jalur 1
File	Stokbarang.php

Lampiran E.19. Update Barang

Pengujian *white box* pada fitur *update* barang adalah sebagai berikut:

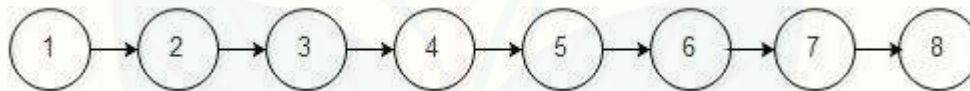
1. Listing Program

```

49 public function update() { 1
50     $id = $this->input->post('kd_barang');
51     $kategori= $this->input->post('kategori'); 2
52     $merk= $this->input->post('merk');
53     $type= $this->input->post('type');
54     $data = array(
55         'kategori' => $kategori,
56         'merk' => $merk,
57         'type'=> $type 3
58     ); 4
59     $where = array(
60         'kd_barang' => $id 5
61     ); 6
62     $this->m_stokbarang->updateBarang($where, $data, 't_barang');
63     redirect('Stokbarang/view');
64 } 7
8
    
```

2. Grafik Alir

Grafik Alir *update* barang. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.19.



Gambar E.19. Diagram Alir *Update* Barang

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *update* barang dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur *update* barang menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3-4-5-6-7-8

5. Test Case

Tabel E.27. *Update* Barang Jalur 1

Test case	User mengubah data barang dan klik tombol
-----------	---

	simpan
Target yang diharapkan	Sistem mengupdate data barang
Hasil pengujian	Benar
Jalur	Jalur 1
File	Stokbarang.php

Lampiran E.20. Hapus Barang

Pengujian *white box* pada hapus barang adalah sebagai berikut:

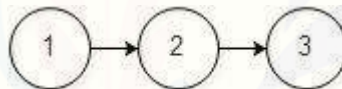
1. Listing Program

```

65 public function delete($id) {} 1
66 { $this->m_stokbarang->DeleteData($id);
67   redirect(base_url('stokbarang/view'));
68 } 3
    
```

2. Grafik Alir

Grafik Alir hapus barang. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.20.



Gambar E.20. Diagram Alir Hapus Barang

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur hapus barang dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur hapus barang menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3

5. Test Case

Tabel E.28. Hapus Barang Jalur 1

Test case	User menekan tombol <i>trash</i>
Target yang diharapkan	Barang terhapus
Hasil pengujian	Benar
Jalur	Jalur 1

File	Stokbarang.php
------	----------------

Lampiran E.21. Menu Merk

Pengujian white box pada fitur menu merk adalah sebagai berikut:

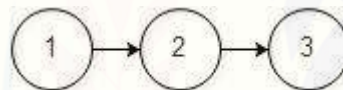
1. Listing Program

```

8 public function view() { } 1
9
10 { $data['merk'] = $this->m_merk->GetMerk();
11   $this->load->view('admingudang/header');
12   $this->load->view('admingudang/merk', $data);
13   $this->load->view('footer');
14 } 2 3
    
```

2. Grafik Alir

Grafik Alir fitur menu merk. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.21.



Gambar E.21. Grafik Alir Menu Merk

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur menu merk dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2-3+2=1$$

4. Basis Set

Basis set dari fitur menu merk menghasilkan 1 jalur yaitu :

$$\text{Jalur1} = 1-2-3$$

5. Test Case

Tabel E.29. Menu Merk Jalur 1

Test case	User menekan menu merk
Target yang diharapkan	Sistem menampilkan halaman merk
Hasil pengujian	Benar
Jalur	Jalur 1
File	Merk.php

Lampiran E.22. Tambah Merk

Pengujian *white box* pada fitur tambah merk adalah sebagai berikut:

1. Listing Program

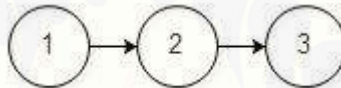
```

15 public function addMerk() { 1
16     $this->load->view('admingudang/header');
17     $this->load->view('admingudang/addmerk'); 2
18     $this->load->view('footer');
19 } 3

```

2. Grafik Alir

Grafik Alir fitur tambah merk. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.22.



Gambar E.22. Grafik Alir Tambah Merk

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur tambah merk dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur tambah merk menghasilkan 1 jalur yaitu :

$$\text{Jalur 1} = 1-2-3$$

5. Test Case

Tabel E.30. Tambah Merk Jalur 1

Test case	User menekan tombol tambah data pada halaman merk
Target yang diharapkan	Sistem menampilkan halaman merk
Hasil pengujian	Benar
Jalur	Jalur 1
File	addmerk.php

Lampiran E.23. *Insert* Merk

Pengujian *white box* pada fitur *insert* merk adalah sebagai berikut:

1. Listing Program

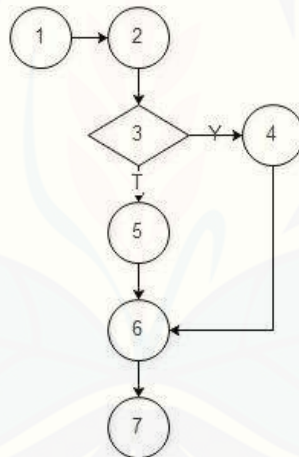
```

20 public function insert(){ } 1
21     $data['kd_merk'] = $this->input->post('kd_merk');
22     $data['merk'] = $this->input->post('merk');
23     $input = $this->m_merk->simpan($data);
24     if($input == true){
25         redirect('merk/view?pesan=berhasil'); } 4
26     }else{
27         redirect('merk/view?pesan=gagal'); } 5
28     } } 6
29 } } 7

```

2. Grafik Alir

Grafik Alir fitur *insert* merk. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.23.



Gambar E.23. Diagram Alir *Insert* Merk

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *insert* merk dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 7 - 7 + 2 = 2$$

4. Basis Set

Basis set dari fitur *insert* merk menghasilkan 2 jalur yaitu :

Jalur 1 = 1-2-3-4-6-7

Jalur 2 = 1-2-3-5-6-7

5. Test Case

Tabel E.31. *Insert Merk Jalur 1*

Test case	User mengisi data merk dengan kode yang belum terdaftar pada sistem dan menekan tombol simpan
Target yang diharapkan	Input data berhasil dan kode tertera dalam tabel
Hasil pengujian	Benar
Jalur	Jalur 1
File	addmerk.php

Tabel E.32. *Insert Merk Jalur 2*

Test case	User mengisi data merk dengan kode yang sudah terdaftar pada sistem dan klik tombol simpan
Target yang diharapkan	Menampilkan halaman merk dan <i>input</i> data tidak tertera pada tabel
Hasil pengujian	Benar
Jalur	Jalur 2
File	addmerk.php

Lampiran E.24. Hapus Merk

Pengujian *white box* pada fitur hapus merk adalah sebagai berikut:

1. Listing Program

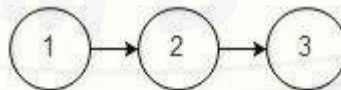
```

30 public function delete($id) { } 1
31     { $this->m_merk->DeleteData($id);
32     } 2
33     } 3

```

2. Grafik Alir

Grafik Alir fitur hapus merk. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.24.



Gambar E.24. Grafik Alir Hapus Merk

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur hapus merk dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. *Basis Set*

Basis set dari fitur hapus merk menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3

5. *Test Case*

Tabel E.33. Hapus Merk Jalur 1

Test case	User menghapus data merk
Target yang diharapkan	Data merk terhapus
Hasil pengujian	Benar
Jalur	Jalur 1
File	Merk.php

Lampiran E.25. Menu Kategori

Pengujian *white box* pada fitur menu kategori adalah sebagai berikut:

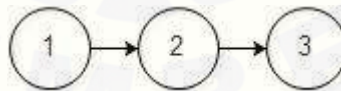
1. *Listing Program*

```

8 public function view() { } 1
9 $data['kategori'] = $this->m_kategori->GetKategori();
10 2 { $this->load->view('admingudang/header');
11 $this->load->view('admingudang/kategori', $data);
12 $this->load->view('footer');
13 } 3
    
```

2. *Grafik Alir*

Grafik Alir fitur menu kategori. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.25.



Gambar E.25. Diagram Alir Menu Kategori

3. *Kompleksitas siklomatis*

Kompleksitas siklomatis dari fitur menu kategori dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. *Basis Set*

Basis set dari fitur menu kategori menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3

5. *Test Case*

Tabel E.34. Menu Kategori Jalur 1

Test case	User menekan menu kategori
Target yang diharapkan	Sistem menampilkan halaman kategori
Hasil pengujian	Benar
Jalur	Jalur 1
File	Kategori.php

Lampiran E.26. Tambah Kategori

Pengujian *white box* pada fitur tambah kategori adalah sebagai berikut:

1. *Listing Program*

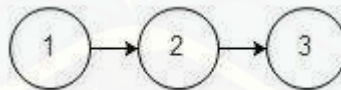
```

15 public function addKategori() {} 1
16 { $this->load->view('admingudang/header');
17 2 { $this->load->view('admingudang/addkategori');
18   { $this->load->view('footer');
19 } } 3

```

2. *Grafik Alir*

Grafik Alir fitur tambah kategori. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.26.



Gambar E.26. Grafik Alir Tambah Kategori

3. *Kompleksitas siklomatis*

Kompleksitas siklomatis dari fitur tambah kategori dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. *Basis Set*

Basis set dari fitur tambah kategori menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3

5. *Test Case*

Tabel E.35. Tambah Kategori Jalur 1

Test case	User menekan tombol tambah data pada halaman kategori
Target yang diharapkan	Sistem menampilkan form kategori
Hasil pengujian	Benar
Jalur	Jalur 1
File	addkategori.php

Lampiran E.27. Insert Kategori

Pengujian *white box* pada fitur *insert* kategori adalah sebagai berikut:

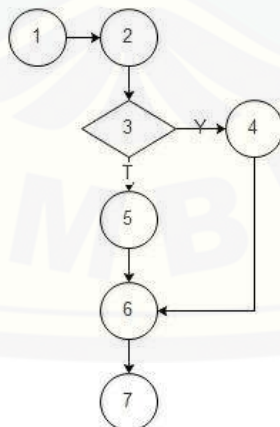
1. Listing Program

```

20 public function insert(){
21     $data['kd_kategori'] = $this->input->post('kd_kategori');
22     $data['kategori'] = $this->input->post('kategori');
23     $input = $this->m_kategori->simpan($data);
24     if($input == true){
25         redirect('kategori/view?pesan=berhasil');
26     }else{
27         redirect('kategori/view?pesan=gagal');
28     }
29 }
    
```

2. Grafik Alir

Grafik Alir fitur *insert* kategori. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.27.



Gambar E.27. Grafik Alir *Insert* Kategori

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *insert* kategori dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 7 - 7 + 2 = 2$$

4. Basis Set

Basis set dari fitur *insert* kategori menghasilkan 2 jalur yaitu :

Jalur 1 = 1-2-3-4-6-7

Jalur 2 = 1-2-3-5-6-7

5. Test Case

Tabel E.36. *Insert* Kategori Jalur 1

Test case	User mengisi data kategori dengan kode yang belum terdaftar pada sistem dan menekan tombol simpan
Target yang diharapkan	<i>Input</i> data berhasil dan kode tertera dalam tabel
Hasil pengujian	Benar
Jalur	Jalur 1
File	addkategori.php

Tabel E.37. *Insert* Kategori Jalur 2

Test case	User mengisi data kategori dengan kode yang sudah terdaftar pada sistem dan klik tombol simpan
Target yang diharapkan	Menampilkan halaman kategori dan <i>input</i> data tidak tertera pada tabel
Hasil pengujian	Benar
Jalur	Jalur 2
File	addkategori.php

Lampiran E.28. Hapus Kategori

Pengujian *white box* pada fitur hapus kategori adalah sebagai berikut:

1. Listing Program

```

30     public function delete($id){ } 1
31     { $this->m_kategori->DeleteData($id);
32     { redirect(base_url('kategori/view'));
33     }
34     } 3

```


2. Grafik Alir

Grafik Alir fitur hapus kategori. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.28.



Gambar E.28. Grafik Alir Hapus Kategori

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur hapus kategori dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur hapus kategori menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3

5. Test Case

Tabel E.38. Hapus Kategori Jalur 1

Test case	<i>User</i> menekan tombol <i>trash</i> pada kolom yang dipilih
Target yang diharapkan	Data kategori yang dipilih terhapus
Hasil pengujian	Benar
Jalur	Jalur 1
File	Kategori.php

Lampiran E.29. Menu Penjualan

Pengujian *white box* pada fitur menu penjualan adalah sebagai berikut:

1. Listing Program

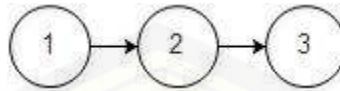
```

7 public function view() { } 1
8     $data['penjualan'] = $this->m_penjualan->GetPenjualan();
9     $this->load->view('adminpenjualan/header');
10    $this->load->view('adminpenjualan/penjualan', $data);
11    $this->load->view('footer');
12 } 3
  
```

2. Grafik Alir

Grafik Alir fitur menu penjualan. Grafik alir tersebut di gambarkan dengan

simpul dan garis dapat di lihat pada Gambar E.29.



Gambar E.29. Grafik Alir Menu Penjualan

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur menu penjualan dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur menu penjualan menghasilkan 1 jalur yaitu :

$$\text{Jalur 1} = 1-2-3$$

5. Test Case

Tabel E.39. Menu Penjualan Jalur 1

Test case	User menekan menu penjualan
Target yang diharapkan	Menampilka halaman data penjualan
Hasil pengujian	Benar
Jalur	Jalur 1
File	Penjualan.php

Lampiran E.30. Tambah Penjualan

Pengujian white box pada fitur tambah penjualan adalah sebagai berikut:

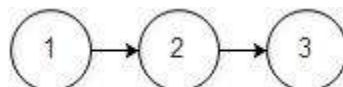
1. Listing Program

```

18 public function addPenjualan() { 1
19     { $this->load->view('adminpenjualan/header');
20     2 { $this->load->view('adminpenjualan/addpenjualan');
21     } $this->load->view('footer');
22 } 3
  
```

2. Grafik Alir

Grafik Alir fitur tambah penjualan. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.30.



Gambar E.30. Grafik Alir Tambah Penjualan

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur tambah penjualan dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur tambah penjualan menghasilkan 1 jalur yaitu :

Jalur 1 = 1-2-3

5. Test Case

Tabel E.40. Tambah Penjualan Jalur 1

Test case	User mengisi form data penjualan
Target yang diharapkan	Form daa penjualan terisi
Hasil pengujian	Benar
Jalur	Jalur 1
File	addpenjualan.php

Lampiran E.31. Insert Penjualan

Pengujian *white box* pada fitur *insert* penjualan adalah sebagai berikut:

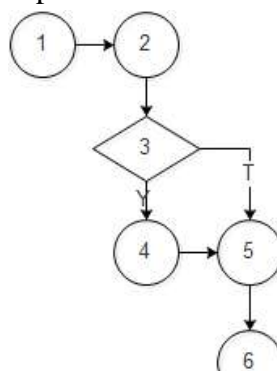
1. Listing Program

```

23 public function insert() { 1
24     $data['kd_penjualan'] = $this->input->post('kd_penjualan');
25     $data['kd_barang'] = $this->input->post('kd_barang');
26     $data['harga_jual'] = $this->input->post('harga_jual');
27     $data['jumlah_jual'] = $this->input->post('jumlah_jual');
28     $input = $this->m_penjualan->simpan($data);
29     if($input == true) { 3
30         redirect('penjualan/view'); 4
31     } 5
32 } 6
    
```

2. Grafik Alir

Grafik Alir fitur *insert* penjualan. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.31.



Gambar E.31. Grafik Alir *Insert* Penjualan

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur *insert* penjualan dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 6 - 6 + 2 = 2$$

4. *Basis Set*

Basis set dari fitur *insert* penjualan menghasilkan 2 jalur yaitu :

Jalur 1= 1-2-3-4-5-6

Jalur 2= 1-2-3-5-6

5. *Test Case*Tabel E.41. *Insert* Penjualan Jalur 1

Test case	User mengisi form data penjualan dan menekan tombol simpan
Target yang diharapkan	Input data berhasil
Hasil pengujian	Benar
Jalur	Jalur 1
File	addpenjualan.php

Tabel E.42. *Insert* Penjualan Jalur 2

Test case	User mengisi form penjualan tidak lengkap dan klik simpan
Target yang diharapkan	Sistem menampilkan " <i>please fill out this field</i> " pada kolom yang belum trisi
Hasil pengujian	Benar
Jalur	Jalur 2
File	addpenjualan.php

Lampiran E.32. Hapus Penjualan

Pengujian *white box* pada fitur hapus penjualan adalah sebagai berikut:

1. Listing Program

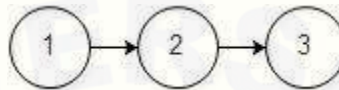
```

33 public function delete($id){ 1
34     $this->m_penjualan->DeleteData($id);
35     redirect(base_url('penjualan/view'));
36 } 2 3

```

2. Grafik Alir

Grafik Alir fitur hapus penjualan. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.32.



Gambar E.32. Grafik Alir Hapus Penjualan

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur hapus penjualan dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. Basis Set

Basis set dari fitur hapus penjualan menghasilkan 1 jalur yaitu :

$$\text{Jalur 1} = 1-2-3$$

5. Test Case

Tabel E.38. Hapus Penjualan Jalur 1

Test case	User menekan tombol trash pada salah satu kolom data penjualan
Target yang diharapkan	Data yang dipilih terhapus
Hasil pengujian	Benar
Jalur	Jalur 1
File	Penjualan.php

Lampiran E.33. Transaksi Barang

Pengujian *white box* pada fitur transaksi data barang adalah sebagai berikut:

1. Listing Program

```

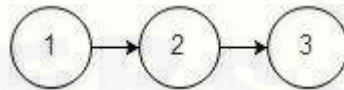
19 }
20 public function owner(){ 1
21     $data['stokbarang'] = $this->m_stokbarang->GetBarangview();
22     $this->load->view('owner/header');
23     $this->load->view('owner/stokbarang', $data);
24     $this->load->view('footer');
25 } 2

```

} 3

2. Grafik Alir

Grafik Alir fitur transaksi barang. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar E.33.



Gambar E.33. Grafik Alir Transaksi Barang

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur transaksi barang dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4. *Basis Set*

Basis set dari fitur transaksi data barang menghasilkan 1 jalur yaitu :

$$\text{Jalur 1} = 1-2-3$$

5. *Test Case*

Tabel E.38. Menu Barang Jalur 1

Test case	<i>User</i> menekan menu barang
Target yang diharapkan	Menampilkan halaman transaksi barang
Hasil pengujian	Benar
Jalur	Jalur 1
File	stokbarang.php

LAMPIRAN F. PENGUJIAN *BLACK BOX***Lampiran F.1. Autentikasi (*login*)**

ALUR NORMAL			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Menjalankan sistem	Menampilkan form <i>login</i>	[✓]	[]
Mengisi <i>username</i> dan <i>password</i> kemudian menekan tombol " <i>login</i> "	Menampilkan halaman sesuai hak akses	[✓]	[]
ALUR ALTERNATIF			
Jika aktor tidak mengisi form <i>login</i> secara lengkap			

Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Menekan tombol <i>login</i> tetapi data tidak diisi dengan lengkap	Menampilkan form <i>login</i>	[✓]	[]
ALUR ALTERNATIF			
Jika aktor salah memasukan <i>username</i> atau <i>password</i>			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Menekan tombol <i>login</i> tetapi <i>password</i> atau <i>username</i> salah	Melakukan validasi <i>username</i> dan <i>password</i> kemudian menampilkan form <i>login</i>	[✓]	[]

Lampiran F.2. Manajemen Data User

ALUR NORMAL			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Memilih Menu “ <i>user</i> ”	Menampilkan halaman data <i>user</i>	[✓]	[]
Menekan tombol “Tambah Data”	Menampilkan form tambah <i>user</i>	[✓]	[]
Mengisi form tambah <i>user</i> dan menekan tombol “Simpan”	Menyimpan data kedalam <i>database</i> dan menampilkan halaman data <i>user</i>	[✓]	[]
Menekan tombol “Trash” pada kolom <i>action</i> di tabel data <i>user</i> sesuai nama <i>user</i> yang ingin dihapus	Menghapus data yang dipilih dan menampilkan halaman data <i>user</i>	[✓]	[]
Menekan tombol “Pencil” pada kolom <i>action</i> di tabel data <i>user</i> sesuai nama <i>user</i> yang ingin diedit	Menampilkan halaman edit data <i>user</i> yang sudah berisi data <i>user</i> yang dipilih	[✓]	[]
Melakukan edit data dan menekan tombol “Simpan”	Melakukan update <i>database</i> kemudian menampilkan halaman data <i>user</i> yang telah terupdate	[✓]	[]

ALUR ALTERNATIF			
Jika aktor menambah <i>user</i> tetapi data yang diinputkan tidak lengkap			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Mengisi form tambah <i>user</i> tetapi data tidak lengkap dan menekan tombol “Simpan”	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi	[✓]	[]
ALUR ALTERNATIF			
Jika aktor ingin mengedit data user tetapi data tidak lengkap			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Melakukan edit data tetapi data yang dimasukkan tidak lengkap dan menekan tombol “Simpan”	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi	[✓]	[]

Lampiran F.3. Manajemen Data Pembelian

ALUR NORMAL			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Memilih Menu “Pembelian”	Menampilkan halaman data pembelian	[✓]	[]
Menekan tombol “Beli Barang”	Menampilkan form tambah pembelian	[✓]	[]
Mengisi form tambah pembelian dan menekan tombol “Simpan”	Menyimpan data kedalam <i>database</i> dan menampilkan halaman data pembelian	[✓]	[]
Menekan tombol “Trash” pada kolom <i>action</i> di tabel data pembelian sesuai nama pembelian yang ingin dihapus	Menghapus data yang dipilih dan menampilkan halaman data pembelian	[✓]	[]
Menekan tombol “Pensil” pada kolom <i>action</i> di tabel data	Menampilkan halaman edit data pembelian yang sudah berisi data pembelian yang	[✓]	[]

pembelian sesuai data yang ingin diedit	dipilih		
Melakukan edit data dan menekan tombol “Simpan”	Melakukan update <i>database</i> kemudian menampilkan halaman data pembelian yang telah terupdate	[✓]	[]
ALUR ALTERNATIF			
Jika aktor menambah pembelian tetapi data yang diinputkan tidak lengkap			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Mengisi form tambah pembelian tetapi data tidak lengkap dan menekan tombol “Simpan”	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi	[✓]	[]
ALUR ALTERNATIF			
Jika aktor ingin mengedit data pembelian tetapi data tidak lengkap			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Melakukan edit data tetapi data yang dimasukan tidak lengkap dan menekan tombol “Simpan”	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi	[✓]	[]

Lampiran F.4. Manajemen Data Barang

ALUR NORMAL			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Memilih Menu “Barang”	Menampilkan halaman data barang	[✓]	[]
Menekan tombol “Tambah Barang”	Menampilkan form tambah barang	[✓]	[]
Mengisi form tambah barang dan menekan tombol “Simpan”	Menyimpan data kedalam <i>database</i> dan menampilkan halaman data barang	[✓]	[]
Menekan tombol	Menghapus data yang	[✓]	[]

“Trash” pada kolom <i>action</i> di tabel data barang sesuai nama barang yang ingin dihapus	dipilih dan menampilkan halaman data barang		
Menekan tombol “Pencil” pada kolom <i>action</i> di tabel data barang sesuai data yang ingin diedit	Menampilkan halaman edit data barang yang sudah berisi data barang yang dipilih	[✓]	[]
Melakukan edit data dan menekan tombol “Simpan”	Melakukan update <i>database</i> kemudian menampilkan halaman data barang yang telah terupdate	[✓]	[]
ALUR ALTERNATIF			
Jika aktor menambah barang tetapi data yang diinputkan tidak lengkap			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Mengisi form tambah barang tetapi data tidak lengkap dan menekan tombol “Simpan”	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi	[✓]	[]
ALUR ALTERNATIF			
Jika aktor ingin mengedit data barang tetapi data tidak lengkap			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Melakukan edit data tetapi data yang dimasukan tidak lengkap dan menekan tombol “Simpan”	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi	[✓]	[]

Lampiran F.5. Manajemen Data Merk

ALUR NORMAL			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Memilih Menu “Merk”	Menampilkan halaman data merk	[✓]	[]

Menekan tombol “Tambah Data”	Menampilkan form tambah merk	[✓]	[]
Mengisi form tambah merk dan menekan tombol “Simpan”	Menyimpan data kedalam <i>database</i> dan menampilkan halaman data merk	[✓]	[]
Menekan tombol “Trash” pada kolom <i>action</i> di tabel data merk sesuai nama merk yang ingin dihapus	Menghapus data yang dipilih dan menampilkan halaman data merk	[✓]	[]
ALUR ALTERNATIF			
Jika aktor menginputkan kode merk yang sudah tersimpan di database			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Mengisi form kode merk dengan kode merk yang sudah ada kemudian menekan tombol “Simpan”	Menampilkan halaman data merk tetapi data yang diinputkan tidak tertera pada tabel	[✓]	[]
ALUR ALTERNATIF			
Jika aktor ingin mengisi data merk tetapi data tidak lengkap			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Melakukan edit data tetapi data yang dimasukan tidak lengkap dan menekan tombol “Simpan”	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi	[✓]	[]

Lampiran F.6. Manajemen Data Kategori

ALUR NORMAL			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Memilih Menu “Kategori”	Menampilkan halaman data kategori	[✓]	[]
Menekan tombol “Tambah Data”	Menampilkan form tambah kategori	[✓]	[]
Mengisi form tambah	Menyimpan data kedalam	[✓]	[]

kategori dan menekan tombol “Simpan”	<i>database</i> dan menampilkan halaman data kategori		
Menekan tombol “Trash” pada kolom <i>action</i> di tabel data kategori sesuai nama kategori yang ingin dihapus	Menghapus data yang dipilih dan menampilkan halaman data kategori	[✓]	[]
ALUR ALTERNATIF			
Jika aktor menginputkan kode kategori yang sudah tersimpan di database			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Mengisi form kode kategori dengan kode kategori yang sudah ada kemudian menekan tombol “Simpan”	Menampilkan halaman data kategori tetapi data yang diinputkan tidak tertera pada tabel	[✓]	[]
ALUR ALTERNATIF			
Jika aktor ingin mengisi data kategori tetapi data tidak lengkap			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Melakukan edit data tetapi data yang dimasukan tidak lengkap dan menekan tombol “Simpan”	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi	[✓]	[]

Lampiran F.7. Manajemen Data Penjualan

ALUR NORMAL			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Memilih Menu “Penjualan”	Menampilkan halaman data penjualan	[]	[]
Menekan tombol “Tambah Data”	Menampilkan form tambah penjualan	[]	[]
Mengisi form tambah penjualan dan menekan tombol “Simpan”	Menyimpan data kedalam <i>database</i> dan menampilkan halaman data penjualan	[]	[]

Menekan tombol “Trash” pada kolom <i>action</i> di tabel data penjualan sesuai nama penjualan yang ingin dihapus	Menghapus data yang dipilih dan menampilkan halaman data penjualan	[]	[]
Menekan tombol “Pencil” pada kolom <i>action</i> di tabel data penjualan sesuai data yang ingin diedit	Menampilkan halaman edit data penjualan yang sudah berisi data barang yang dipilih	[]	[]
Melakukan edit data dan menekan tombol “Simpan”	Melakukan update <i>database</i> kemudian menampilkan halaman data penjualan yang telah terupdate	[]	[]
ALUR ALTERNATIF			
Jika aktor menambah data penjualan tetapi data yang diinputkan tidak lengkap			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Mengisi form tambah penjualan tetapi data tidak lengkap dan menekan tombol “Simpan”	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi	[]	[]
ALUR ALTERNATIF			
Jika aktor ingin mengedit data penjualan tetapi data tidak lengkap			
Aktor	Sistem	STATUS	
		SUKSES	GAGAL
Melakukan edit data tetapi data yang dimasukan tidak lengkap dan menekan tombol “Simpan”	Menampilkan pesan “ <i>Please fill out this field</i> ” pada form yang belum terisi	[]	[]

Lampiran F.8. Manajemen Data Transaksi

ALUR NORMAL		
Aktor	Sistem	STATUS

			SUKSES	GAGAL
Memilih “Barang”	Menu	Menampilkan halaman data barang	[✓]	[]
Memilih Menu “User”		Menampilkan halaman data <i>user</i>	[✓]	[]
Memilih “Pembelian”	Menu	Menampilkan halaman data pembelian	[✓]	[]

LAMPIRAN G. DATASET

Lampiran G. Data penjualan barang di CV.Unicorner

Bln/Thn/Tgl	Kategori	Nama Barang	Qty	
Desember 2014	5	Acc	Charger Oppo	1
			Flashdisk Toshiba 8 Gb	1
			Flashdisk Toshiba 8 Gb	2

	8		Memori Sandisk class10 8 Gb	1
	9		Headset Karakter	1
	10		Kabel data	1
			Flashdisk 16 Gb	1
	12		Flashdisk 8 Gb	3
	16		Charger Asus	1
			Charger Samsung	1
			Kabel data micro USB	2
	18		Headset Samsung	1
	20		Flashdisk 8 Gb	2
			Headset Bluetooth	1
	23		Headset Iphone	1
	30		Kabel data Iphone 3S	1
Januari 2015	6		Memori Sandisk 8 Gb	1
			Cassing Oppo	1
			Cassing Samsung	1
			Iphone5 16 Gb	1
	10		Earpod	1
			Headset Karakter	2
	13		Headset Karakter	1
	14		Charger Oppo 2A	1
	15		Charger Iphone5	1

	20		Case Oppo	1
			Headset Samsung	1
	23		Flashdisk 8 Gb	2
			Flashdisk Toshiba 8 Gb	1
	28		Memori Sandisk 8 Gb	1
Februari 2015	2		Flashdisk Toshiba 16 Gb	1
	4		Flashdisk Toshiba 8 Gb	1
			Headset Samsung	1
	10		Cas Asus	1
	11		Headset Iphone	1
	13		Headset Karakter	2
			Cassing Transparan	1
			Case 360 TG Samsung	1
	16		Charger Samsung	1
			Charger Oppo	1
			Charger Iphone5S	1
	18		Charger Samsung	1
			Kabel data Samsung	1
	23		Kabel data Iphone4	1
			Powerbank Hippo	1
			Memori Sandisk 8 Gb	1
	25		Iphone5 16 Gb	1

			Charger Samsung	1
	26		Flashdisk 8 Gb	1
	28		Iphone5 64 Gb	1
Maret 2015	2		Tongsis	1
	3		Charger Asus	1
			Car charger Samsung dual port	1
	5		Kabel data Ipad	1
	6		Flashdisk Toshiba	1
			Charger Oppo 2A	1
	10		Kabel data Samsung	1
			Flashdisk 8 Gb	2
	11		Flashdisk 8 Gb	3
	12		Kabel data Oppo	1
	13		Headset Karakter	1
			Kabel data Samsung	1
	17		Kabel data Ipad	1
			Memori Sandisk	1
			Headset Samsung	1
	19		Headset Sony mdr xb450	1
	20		Headset Iphone	1
23		Charger Oppo	1	
25		Iphone5 64 Gb	1	

	26		Powerbank Hippo	1
April 2015	1		Headset Samsung	1
	6		Headset Iphone	1
			Iphone 6 16 Gb	1
			Flashdisk 8 Gb	1
	7		Memori Sandisk 32 Gb	1
			Flashdisk 8 Gb	1
	8		Iphone 6S 64 Gb	1
			Waterproof case	3
	9		Cassing transparan Iphone	2
			Charger Oppo 2A	1
	10		Kabel data Samsung	1
			Flashdisk 8 Gb	1
	13		Flashdisk Toshiba 8 Gb	1
	14		Kabel data Iphone5	1
	21		Charger Iphone5S	1
30		Charger Asus	1	
Mei 2015	4		Kabel data Ipad2	1
			Headset Bluetooth	1
			Iphone5 64Gb	1
	6		Headset Samsung	1
			Earpod Iphone	1

	7		Iphone6Plus 64 Gb	1
			Memori Sandisk 32 Gb	1
			Cassing Transparan	1
	8		Waterproof case mobile phone bag	1
			Kabel data micro USB	1
	11		Flashdisk 32 Gb	1
	13		Flashdisk 8 Gb	2
			Headset Karakter	1
	18		Flashdisk Toshiba 8 Gb	3
			Kabel data Samsung	1
	20		Charger Asus	1
	22		Charger Oppo 2A	1
			Cassing Oppo	1
	25		Kabel data Samsung	1
	29		Headset Karakter	1
			Flashdisk 8 Gb	1
Juni 2015	3		Iphone6 64 Gb	1
			Headset Karakter	2
	4		Flashdisk 8 Gb	1
			Kabel data Samsung	1
	5		Iphone5 64 Gb	1

		Flashdisk 8 Gb	1	
	9	Headset Iphone	1	
	11	Memori Sandisk 32 Gb	1	
		Headset Bluetooth	1	
	17	Iphone6 16 Gb	1	
	19	Charger Oppo 2A	1	
	22	Case 360 TG Samsung	1	
		Cassing anti crack	1	
	24	Cassing Transparan Iphone6	1	
Juli 2015	1	Charger iphone 5S	1	
		Kabel data Iphone3	1	
	2	Headset Karakter	1	
		Flashdisk 8 Gb	1	
		Memori Sandisk 32 Gb	1	
	3	Flashdisk 32 Gb	1	
		Charger Asus	1	
	6	Iphone6Plus 64 Gb	1	
		Memori Sandisk 8 Gb	1	
	9	Charger Iphone6	1	
	13	Headset Iphone	1	
	15	Iphone5 64 Gb	1	
			Headset Samsung	1

	27		Flashdisk 8 Gb	1
			Cas Asus	1
	30		Charger Samsung dual port	1
Agustus 2015	3		Iphone6 16 Gb	1
			Charger Oppo	1
	5		Cassing Transparan Iphone6	1
			Waterproof case mobile phone bag	4
	6		Cassing anti crack Iphone	1
			Charger Asus	1
			Kabel data micro USB	1
			Fashdisk 32 Gb	1
	12		Headset Karakter	1
	14		Memori Sandisk 8 Gb	1
			Headset Samsung	1
	20		Charger Oppo	1
			Headset Karakter	2
	25		Earpod Iphone	1
		Iphone6 64 Gb	1	
September 2015	1		Kabel data Iphone4	1
			Headset Samsung	1
	3		Kabel data Samsung	1

	4	Cassing Transparan Iphone6	1
		Waterproof case mobile phone bag	5
	7	Charger Oppo 2A	1
		Cassing Oppo	1
	9	Kabel data Samsung	1
		Charger Asus	1
	10	Headset Samsung	1
	11	Memori Sandisk 8 Gb	1
		Memori Sandisk 32 Gb	1
		Headset Karakter	2
	15	Charger Iphone5S	1
		Charger Samsung	1
	18	Cassing Transparan	1
		Earpod Iphone	1
	22	Kabel data Iphone	1
		Cassing anti crack	1
	29	Headset Sony mdr xb450	1
		Kabel data micro USB	1
Oktober 2015	1	Flashdisk Toshiba 8 Gb	1
		Flashdisk Toshiba 8 Gb	2
	2	Iphone6Plus 64 Gb	1

		Iphone6S cover	1
	6	Kabel data Samsung	1
		Memori Sandisk 32 Gb	1
	12	Charger Iphone6	1
		Powerbank Hippo	1
	16	Cassing anti crack	1
		Headset Karakter	1
	21	Headset Samsung	2
Nopember 2015	2	Kabel data Samsung	1
		Headset Samsung	1
	4	Headset Iphone	1
		Kabel data micro USB	1
	10	Charger Oppo	1
		Charger Oppo 2A	1
	16	Case TG samsung	1
	20	Cas Asus	1
		Memori Sadisk 8 Gb	1
		Earpod Ipod	1
		Waterproof case	1
		Case Oppo	1
		Charger Oppo 2A	1
	23	Kabel data micro USB	1

			Headset Bluetooth	1
			Headset Karakter	1
	24		Headset Karakter	1
			Headset Samsung	1
	25		Kabel data Samsung	1
	26		Waterproof case	3
	27		Headset Iphone	1
			Cassing Transparan	1
			Charger Samsung dual port	1
	30		Memori Sandisk 8 Gb	1
			Charger Asus	1
			Headset Samsung	1
Desember 2015	1		Flashdisk 8 Gb	3
			Waterproof case mobile phone bag	1
			Waterproof case mobile phone bag	4
	2		Charger Oppo 2A	1
			Iphone6Plus 16 Gb	1
			Flashdisk Toshiba 8 Gb	1
	3		Flashdisk Toshiba 8 Gb	2
	4		Waterproof case mobile phone bag	2

		Charger Oppo	1
	8	Cassing Transparan Iphone6	1
	11	Memori Sandisk 8 Gb	1
		Kabel data Iphone3	1
	14	Iphone 6Plus 64 Gb	1
		Flashdisk 8 Gb	2
		Flashdisk 8 Gb	1
	15	Waterproof case mobile phone bag	3
	18	Flashdisk Toshiba 8 Gb	1
	21	Cassing Transparan	1
		Kabel data Samsung	1
	28	Headset Karakter	1
	29	Headset Samsung	1
	30	Kabel data Iphone4	1